Contributions to Mechanisms for Adaptive Use of Mobile Network Resources

Olivier Mehani

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n°432 :
SMI—Sciences des métiers de l'ingénieur

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris

en cotutelle avec
School of Electrical Engineering and Telecommunications
University of New South Wales, Australie

Spécialité « Informatique temps-réel, robotique et automatique »

présentée et soutenue publiquement par

Olivier Mehani

le 14 décembre 2011

Contributions aux mécanismes de réseau pour un usage adaptatif des ressources mobiles

Direction de la thèse: **Roksana Boreli**
Arnaud de La Fortelle
Encadrement de la thèse: **Thierry Ernst**

Jury

Christian Bonnet, Professeur, EURECOM	Rapporteur
Roksana Boreli, Chercheuse, NICTA	Examinatrice
Thierry Ernst, Chercheur, Mines ParisTech	Examinateur
Arnaud de La Fortelle, Professeur, Mines ParisTech	Examinateur
François Spies, Professeur, Université de Franche-Comté	Président
Andreas Timm-Giel, Professeur, Hamburg University of Technology	Rapporteur

MINES ParisTech
Dép. Mathématiques et Systèmes—Centre de Robotique
60, Boulevard Saint-Michel 75272 PARIS cedex 06

T H È S E

Contributions to Mechanisms for Adaptive Use of Mobile Network Resources

Olivier Mehani

A dissertation submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy

The School of Electrical Engineering and Telecommunications
The University of New South Wales

in a cotutelle agreement with

Dép. Mathématiques et Systèmes—Centre de Robotique
École Nationale Supérieure des Mines–ParisTech, France

14 December 2011

Acknowledgements

I would first like to thank my supervisors, Roksana Boreli, Thierry Ernst, Arnaud de La Fortelle and Aruna Seneviratne for their support throughout my work on this thesis. Roksana has been a great help while refining and expressing the ideas presented in this thesis, while Thierry's input put them into different and wider perspectives. Arnaud accepted to supervise me for Mines–ParisTech which allowed me to work in this international collaboration context, and Aruna was a strong support at NICTA without whom coming to Australia and securing (and extending) my scholarship would have been much harder.

I am grateful to NICTA and Inria, which have financially supported my doctoral studies, and provided a most adequate work environment throughout my research. I am also very thankful to Christine at École des Mines, Victoria at UNSW, Rema at NICTA and Chantal at Inria for their help making all administrative hurdles vanish almost magically.

This thesis has benefited at all stages from my interactions with many people. I am extremely grateful to Guillaume J. who has been a constant support in many ways (ideas, advice, proofreading and beers) throughout these four years. It was also very insightful to work on research papers with my various co-authors (amongst others and in no particular order, Jolyon, Michael and Thierry R.). Rodrigo's advice, even before I started my thesis, has helped me structure my work into something hopefully more coherent. Finally, I extend my gratitude to all the other people I had the opportunity to meet, both in NICTA's Network Research Group (particularly those who willingly exposed themselves to proofreading drafts of this dissertation: Guillaume S., Babil, Yan and Mentari) and Inria's Imara project-team (Armand, Laurent, François, Clément) for making these labs exciting places to work, learn and develop new ideas.

Finally, I would like to thank my parents, who have always been very supportive, even when some decisions did not seem to be the most optimal at the time, or when some others would take me geographically further and further away from them. I also thank Jenny, for her constant support, understanding, and everything else. Moreover, without her, the word "quintessential" would not have appeared in this dissertation.

Abstract

With the widespread availability of multiple wireless network technologies, mobile computing devices can benefit from almost uninterrupted connectivity by changing network attachments as they move. This however raises the problem of the selection method to be used for the choice of the wireless networks to associate with, in order to provide the best performance. Moreover, mobility events may result in poor application quality, due to either a disruption in connectivity during the handover or the heterogeneity of the characteristic of different access networks. To address these problems, this thesis introduces and studies all three elements (observation, decision, action) of a control framework to enable better use of available network resources.

We first show that a decision mechanism which directly considers the relevant user- and application-centric metrics is more appropriate than using the common network metrics-based indirect approach. This mechanism is used to control the entire network stack of the mobile node in a coordinated way, rather than individual components, to avoid potentially conflicting combinations. Our results indicate that, by exploiting the flexibility of application parameters, it is possible to maintain high application quality while reducing both the power consumption and access price.

We then introduce a mobility-aware extension to the TCP-Friendly Rate Control mechanism (TFRC), as an action element, to address the disruption in connectivity resulting from the mobility events. We propose to suspend the transmission before disconnections and to probe the network after reconnections. Simulations demonstrate how this enables faster recovery after disconnected periods as well as a significantly improved adaptation to the newly available network conditions. When used with the Datagram Congestion Control Protocol (DCCP), experiments show that it provides better support for real-time applications for which the user-perceived quality is very dependent on the immediate transmission rate.

Finally, we present an experimental process to evaluate the OMF Measurement Library (OML), a lightweight instrumentation and reporting tool which we propose to use as the observation element of our framework. We show that this library does not significantly impact the performance of the instrumented applications, while accurately reporting the observed metrics.

Keywords network mobility, communication stack, cross-layer framework, optimisation, quality of experience, transport protocol.

Résumé

Avec les larges déploiements de multiples technologies sans fil, les terminaux informatiques mobiles bénéficient d'une connectivité presque permanente, changeant de réseaux d'accès au gré de leurs déplacements. Ceci pose cependant le problème de la sélection de ces réseaux, afin de fournir les meilleures performances. Cette mobilité risque aussi d'impacter la qualité des applications, souvent lors des « handovers » d'un réseau à l'autre, ou en raison de la disparité des caractéristiques des réseaux d'accès. Pour aborder ces problèmes, cette thèse introduit et évalue trois éléments de contrôle (observation, décision, action) permettant une meilleure utilisation des ressources réseau par les équipements mobiles.

Nous montrons d'abord qu'un mécanisme de décision qui utilise directement les métriques pertinentes pour les utilisateurs et les applications est plus approprié que l'approche indirecte classique basée sur les métriques réseau. Ce mécanisme contrôle de manière coordonnée l'ensemble de la pile protocolaire, plutôt que des composants séparés, afin d'éviter des combinaisons conflictuelles. Nous démontrons que la flexibilité des paramètres applicatifs peut être exploitée et permet de maintenir une qualité élevée pour les applications tout en réduisant les coûts d'accès (énergétique et financier).

Une extension au *TCP-Friendly Rate Control mechanism* (TFRC) est ensuite introduite, en tant qu'élément d'action, pour atténuer les perturbations lors des handovers. Nous proposons de suspendre la transmission avant la déconnexion, puis de sonder le réseau après reconnexion. Nous montrons que cela permet un rétablissement plus rapide et une meilleure adaptabilité aux conditions du nouveau réseau. Son usage en combinaison avec le *Datagram Congestion Control Protocol* (DCCP) offre un meilleur support aux applications temps-réel, dont la qualité dépend de la vitesse de transmission immédiate.

Finalement, nous présentons une méthode pour évaluer la *OMF Measurement Library* (OML), une bibliothèque d'instrumentation dont nous proposons l'usage comme élément d'observation. Nous montrons que cette bibliothèque n'a pas d'impact significatif sur les applications instrumentées et qu'elle permet un suivi précis des métriques idoines.

Mots clés mobilité réseau, pile protocolaire, architecture inter-couche, optimisation, qualité d'expérience, protocole de transport.

Contents

Résumé de la thèse en français

1 Introduction (chapitre 1)

1.1 Anciens modèles et le Nouvel Internet

Les protocoles supportant l'Internet à l'heure actuelle sont pour la plupart les mêmes qu'à l'origine (Leiner *et al.*, 2009). Suivant une structure en couches abstraites, fournissant chacune une fonctionnalité bien définie (Zimmermann, 1980), la pile protocolaire TCP/IP a réussi à bien s'adapter aux évolutions des technologies sous-jacentes. Cependant ces nouvelles technologies ont graduellement étendu le contexte dans lequel ces protocoles doivent fonctionner.

Ainsi, la récente *prévalence des réseaux d'accès sans-fil* présente de nouveaux problèmes à des protocoles dont les hypothèses de conception sont basées sur l'usage d'un réseau filaire fiable. Le classique protocole de transport TCP, par exemple, souffre d'une mauvaise interprétation des pertes de paquets dues aux conditions du medium radio et non à des congestions sur le réseau. Il ne peut cependant pas distinguer la cause de ces pertes en raison de la séparation en couches. Cette erreur entraîne des baisses de performances lorsque des réseaux sans fil sont traversés (Xylomenos *et al.*, 2001).

La réduction en taille et en prix des équipements informatiques a largement facilité l'apparition de *l'informatique mobile et l'accès simultané à plusieurs réseaux.* Ces équipements informatiques miniaturisés—intelliphones, tablettes et autres ordinateurs portables—sont transportés au rythme des déplacements de leur utilisateur. Plus récemment, ces terminaux se sont aussi vus pourvoir de plus d'une interface réseau. Ainsi, ils peuvent, au gré des mouvements de leur porteur, établir une ou plusieurs connexions à de multiples réseaux afin d'être « toujours connectés au mieux » (ABC ; Gustafsson et Jonsson, 2003).

De *nouveaux modes de connectivité* ont aussi fait leur apparition. Les réseaux mobiles *ad hoc* (MANET ; Perkins et Royer, 1999; Jacquet *et al.*, 2002), tolérants aux délais (DTN ; Fall, 2003) ou fournis par d'autres utilisateurs (UPN ; Sofia et Mendes, 2008) étendent encore le choix de connectivités parmi lesquelles les plus adéquates doivent être sélectionnées.

Enfin, l'explosion du nombre d'équipements, de plus en plus souvent portables ou embarqués, implantant une pile protocolaire utilisable sur Internet pose un *problème de passage à l'échelle.* Le protocole d'adressage le plus répandu, IPv4, est en passe d'arriver à court d'adresses (Huston, 2011). Ainsi, une nouvelle version, IPv6, a été proposée, et est en cours de déploiement pour apporter une solution à ce problème.

Bien que le problème de passage à l'échelle soit déjà résolu par IPv6, le reste des problèmes introduits par les accès sans fil et la mobilité sont toujours d'actualité. De plus, de nouveaux cas d'usage tirant parti de ces nouvelles opportunités ont émergé, introduisant de nouvelles contraintes et demandes pour la pile TCP/IP.

1.2 Cas d'usage émergents

Auparavant utilisés pour quelques tâches bien spécifiques telles que l'accès distant ou le courriel, les réseaux informatiques ont vu de nouvelles applications émerger en parallèle à l'augmentation de la puissance des machines les faisant tourner. Les *applications multimédia* telles que la vidéo à la demande (VoD) sont maintenant la norme dans l'usage quotidien d'Internet. Ce type de contenu multimédia place une contrainte grandissante sur la capacité des réseaux les transportant, qu'ils soient filaires ou sans fil. De plus, d'autres paramètres tels que les délais de transmission de bout-en-bout doivent aussi être maintenus sous contrôle pour assurer une lecture sans heurt. Une certaine flexibilité peut être obtenue par l'usage de mémoires tampons ; cependant, des applications interactives bidirectionnelles, telles que la voix sur IP (VoIP) ou la vidéo conférence, ne peuvent composer avec les délais additionnels introduits par ces tampons. D'autres solutions doivent donc être étudiées.

Le domaine des *systèmes de transport intelligents* (ITS) connaît aussi un essor d'intérêt de la part de la communauté scientifique (Laurgeau, 2009). Les standards en cours de développement à l'ETSI et l'ISO spécifient l'usage d'IPv6 et ses extensions (ETSI EN 302 665, 2010; ISO 21210:2011) pour les communications véhiculaires. Habiliter ces véhicules à communiquer sur des canaux standards permet d'imaginer une collaboration à grande échelle entre eux et avec l'infrastructure de transport. Plusieurs classes d'applications sont envisagées (Khaled *et al.*, 2009) : sûreté routière, gestion de trafic et de flotte, et info-divertissement. L'info-divertissement est un sur-ensemble des applications multimédia, et partage les mêmes besoins, alors que les applications de gestion et de sûreté introduisent des contraintes plus fortes, mais pouvant être supportées de manière similaire.

1.3 Besoin de solutions palliatives

Plusieurs problèmes dus à la disparité entre ce que la pile protocolaire TCP/IP peut supporter et les nouveaux cas d'usage se posent donc :

- question du choix de(s) réseau(x) d'accès le(s) plus adapté(s) pour supporter au mieux les flots applicatifs courants ;
- impossibilité des algorithmes implantés dans la pile à s'adapter rapidement à des changements de conditions, résultant en un service non optimal fourni aux applications.

Dans cette thèse, nous nous proposons d'aborder ces problèmes en apportant des éléments de réponse à la question suivante :

> Comment permettre à des nœuds mobiles de maintenir la meilleure qualité applicative possible en utilisant des ressources réseau changeant dynamiquement, et supporter une dégradation contrôlée de la qualité quand ces ressources sont insuffisantes ?

Plusieurs types de solutions ont déjà été proposés pour régler les problèmes mentionnés ci-dessus. Certains proposent de se débarrasser de la structure en couche de TCP/IP, et de reconcevoir une architecture protocolaire plus adaptée (p. ex. Clark, 2009). Cette approche pose cependant le problème de ne pas être rétro-compatible avec les solutions existantes, standardisées et déployées, et risque de poser des problèmes de migration. Il semble donc plus raisonnable de proposer des améliorations incrémentales et graduellement se rapprocher d'une structure mieux adaptée aux besoins et contextes actuels et futurs. De nombreuses approches inter-couche, étendant l'interface entre les éléments de la pile protocolaire pour permettre un échange plus riche d'informations, ont été proposées (Srivastava et Motani, 2005), mais des doutes subsistent quant à leur applicabilité (Kawadia et Kumar, 2005).

Afin d'assurer la possibilité d'une transition graduelle, nous préférons revisiter cette seconde approche, tout en essayant d'en lever les limites d'applicabilité. Dans ce but, nous suggérons de suivre le modèle de la boucle d'Observation, Orientation, Décision et Action (OODA ; fig. 1.1, page 6) introduite dans l'armée états-unienne pour s'adapter à un environnement changeant (Boyd, 1995), et de clairement en séparer les trois phases principales pour éviter les interactions conflictuelles.

1.4 Contributions de cette thèse

Dans cette thèse, nous proposons une architecture inter-couche découplant les trois phases principales d'observation, décision et action de la boucle OODA. Les contributions présentées dans cette dissertation sont plusieurs éléments de cette architecture.

Le chapitre 3 introduit un système de décision résolvant le *problème de gestion des flots dans un environnement multi-connecté*. Ce chapitre présente et modélise le problème, puis démontre que prendre en compte les critères perceptibles par l'utilisateur ou pertinents pour les applications permet de sélectionner les réseaux d'accès, la répartition des flots et les paramètres applicatifs et de transport de manière plus adéquate en terme de qualité applicative qu'en considérant uniquement la qualité de service (QoS) offerte par les-dits réseaux.

Le chapitre 4 propose d'abord un modèle du comportement de TFRC lors des changements de réseau. Il introduit ensuite une modification de l'algorithme de contrôle de vitesse d'émission afin qu'il puisse réagir correctement aux événements de mobilité provoquant des déconnexions temporaires. Utilisé au sein de notre architecture, il peut être prévenu en avance de ces déconnexions, et se réadapter rapidement aux caractéristiques du nouveau chemin réseau par lequel le flot est transmis.

Enfin, le chapitre 5 étudie OML, une bibliothèque d'instrumentation et de rapport, en vue de son intégration dans notre architecture comme élément d'observation. Basé sur une large quantité d'expériences, nous quantifions le biais introduit par l'instrumentation dans les mesures et les performances des applications instrumentées. Dans la plupart des cas, aucune différence significative n'est identifiée et, dans les cas où une différence est effectivement confirmée, des pistes sont étudiées pour la réduire.

L'ensemble des contributions—publications et logiciel—provenant des travaux présentés dans cette dissertation est listé en détails dans l'appendice A (page 161).

2 Contexte et état de l'art (chapitre 2)

Le travail présenté dans cette thèse se base sur les réseaux IP, et propose des améliorations incrémentales pour faire face aux nouveaux contextes technologiques et aux nouveaux besoins applicatifs. Le chapitre 2 commence donc par donner une vue d'ensemble du contexte actuel de ces réseaux, avec une emphase sur le support de la mobilité.

Ainsi, les couches principales de la pile protocolaire sont présentées. La couche réseau (au niveau 3), instanciée par IP, se trouve à la taille du sablier représentant cette pile (fig. 2.1, page 13). De nombreux protocoles de transport (de niveau 4 dans l'architecture OSI, comme TCP, UDP, DCCP ou SCTP) reposent sur la capacité d'IP à interfacer d'aussi nombreux media de communication (telles qu'ethernet, Wi-Fi, 3G ou WiMAX). Nous y notons, à la couche réseau, l'urgent besoin de migration à IPv6, l'espace d'adressage d'IPv4 étant épuisé à l'IANA depuis février 2010. Par la suite, les différences sémantiques principales des protocoles de transport y sont classifiées : *fiabilité* pour TCP et SCTP, *contrôle de congestion* pour TCP, DCCP (avec, p. ex., TFRC) et SCTP et *support de datagrammes* pour UDP, DCCP et SCTP. Les caractéristiques des principales technologies utilisées pour les couches physique et de liaison (niveaux 1–2) sont aussi comparées, avec un accent sur les technologies sans fil. La table 2.2 (page 16) recense les principales caractéristiques, à la fois définies par les standards idoines et mesurées sur de réels déploiements, de ces technologies.

L'impact de ces nouvelles implémentations des couches basses (niveaux 1–2) sur les couches hautes (principalement au niveau 4) est présenté plus en détails, ainsi que les divers problèmes introduits par la mobilité des nœuds. Ces problèmes peuvent être liés aux performances des protocoles de transport dont certaines hypothèses de conception sont violées sur les réseaux sans fil (p. ex., pertes de paquets dues à des problèmes transitoires sur le lien radio, plutôt qu'à un chemin congestionné ; voir, p. ex., Xylomenos *et al.*, 2001). La mobilité cause d'autres problèmes de performance, comme les différences de caractéristiques entre les technologies d'accès, auxquels les algorithmes de contrôle de congestion traditionnels ne sont pas adaptés (Bansal *et al.*, 2001). Un problème sémantique est aussi introduit par la mobilité, en ce que les adresses réseau, qui sont utilisées indifféremment et conjointement en tant qu'identifiant et localisateur sur un réseau fixe, voient leurs sens séparés dans un réseau mobile ; les protocoles des couches hautes, conçus pour ne manipuler qu'une seule adresse, ont donc des problèmes pour maintenir leur sessions. Finalement, les possibilités de multi-connexion sont rarement exploitées au mieux, et des choix sous-optimaux sont souvent privilégiés (Wasserman et Seite, 2011).

Plusieurs systèmes de mobilité, de la couche réseau à la couche applicative (niveau 7 dans le modèle OSI), ont été proposés pour régler le problème sémantique de l'identifiant–

localisateur (Nazir et Seneviratne, 2007). Au niveau réseau, il s'agit pour la plupart de mettre à jour de manière transparente le(s) localisateur(s) correspondant à un identifiant particulier, et n'utiliser que ce dernier dans les couches hautes. Plusieurs mécanismes de micro-mobilité étendent ces solutions afin de réduire la surcharge due à la mise à jour de ces associations. Dans les couches les plus élevées, la mobilité est gérée de manière plus *ad hoc* par l'adjonction de protocoles permettant d'identifier une session et de la continuer depuis une nouvelle adresse. De par son ubiquité comme interface entre les couches hautes et basses, il semble cependant que la couche réseau est la plus à même de supporter un déploiement massif de fonctionnalités de mobilité, sans pour autant nécessiter une réécriture des couches hautes au cas par cas. De fait, nous concluons que l'usage de Mobile IPv6 (MIPv6) et de ses extensions (résumés en fig. 2.2, page 22) est le plus adapté pour supporter les améliorations incrémentales que nous proposons.

Pour réduire ou régler les problèmes de performance, de nombreuses interactions inter-couche ont été proposées, à tous les niveaux de la pile protocolaire. Celles ci sont donc passées en revue, et classifiées selon les modèles proposés par Srivastava et Motani (2005 ; fig. 2.3, page 24 et 2.4, page 25). Plusieurs problèmes existent cependant avec ces optimisations inter-couche. La plupart des propositions sont souvent attachées à un problème bien spécifique et, quand une implémentation réelle existe, la solution est rarement testée dans des cas plus génériques. Kawadia et Kumar (2005) observent ainsi que ces approches risquent de violer les hypothèses des protocoles non impliqués dans l'optimisation, et les performances résultantes de se voir dégradées plutôt qu'améliorées. Nous concluons qu'un système inter-couche doit, pour être viable, découpler décision et action, ne laissant que cette dernière phase sous le contrôle du protocole et sur commande, par le biais de telles interfaces, du système de décision. Allant dans ce sens, certains organismes de standardisation ont d'ailleurs commencé à introduire des possibilités d'interaction inter-couche basées sur une API externe (Teraoka *et al.*, 2008; ISO/CD 24102:2008; IEEE Std 802.21-2008), mais se sont effectivement limités à décrire une interface de communication et non un système de décision.

Nous passons donc en revue les divers systèmes de décisions proposés dans la littérature pour sélectionner les réseaux d'accès et y distribuer les flots applicatifs dans le but d'obtenir la meilleure connexion (Gustafsson et Jonsson, 2003). Nous identifions les divers critères selon lesquels les réseaux sont discriminés tels que la puissance du signal, la QoS du lien ou du chemin réseau ainsi établi, la consommation électrique induite par l'usage de l'interface réseau idoine, le prix d'usage, ou encore d'autres critères indirectement liés au réseau. Nous notons aussi certains travaux qui, plutôt que mesurer ces critères, tentent de les estimer par d'autres moyens tels que l'apprentissage automatique ou la mesure collaborative. Certaines des approches mentionnées abordent le problème en considérant plusieurs critères. L'utilisation de fonctions d'utilité basées sur des combinaisons linéaires des critères considérés est souvent la méthode employée pour ce faire. Cependant, d'autres méthodes d'optimisation multiobjectif plus élaborées sont aussi utilisées, comme l'optimisation linéaire, les distances de similarité ou les vraisemblances statistiques.

Le problème de distribution des flots applicatifs dans les environnements multi-connectés repose souvent sur les mêmes critères que la sélection de réseaux, mais deux approches se distinguent. La première consiste à aborder le problème, de manière séparée, avec des techniques de répartition de charge sur les liens établis. La seconde, plus globale, essaie de résoudre conjointement les problèmes de sélection de réseau et de distribution de flots. Compte tenu de la possibilité d'interagir avec l'ensemble des couches de la pile protocolaire, il semble cependant plus indiqué de proposer un système de décision qui puisse prendre en compte tous les éléments et protocoles impliqués, et parvenir à une solution couvrant l'ensemble de leurs paramètres.

Finalement, nous proposons un rapide examen des métriques[1] pertinentes pour évaluer la QoS des réseaux ainsi que la qualité d'expérience (QoE) des applications considérées (Stankiewicz *et al.*, 2011). Pour la première, nous utilisons les définitions de l'IETF, qui propose un cadre cohérent pour l'interprétation et l'échantillonnage des capacités, délais et autres mesures dérivées (Paxson *et al.*, 1998). Pour la seconde, l'ITU a fait un large effort de définition et d'expérimentation ayant mené au développement du MOS, une échelle de qualité perçue entre 1 et 5, ainsi que son expression sous forme d'un modèle objectif pour les conversations audio et vidéo ainsi que les sessions de navigation sur la toile. Nous complétons cette revue avec la définition du PSNR qui permet de quantifier la dégradation d'une (série d')image(s) par comparaison avec l'originale.

Pour ces métriques qui peuvent être directement mesurées (c.-à-d. celles n'impliquant pas l'avis d'un être humain), nous poursuivons avec une revue des outils pour ce faire. Nous notons l'existence de multiples outils dédiés, mais dont les formats incompatibles ne permettent pas une consolidation aisée des données provenant de plusieurs sources. Nous nous penchons donc par la suite sur les protocoles et bibliothèques permettant de collecter et d'agréger ces mesures hétérogènes dans un format unique. Parmi ceux ci, OML se démarque par son API permettant l'instrumentation d'outils existants, ses possibilités de rapport à distance et de traitement en ligne ainsi que sa présumée légèreté.

3 Optimisation inter-couche des choix et usages des réseaux d'accès (chapitre 3)

Les terminaux mobiles, qu'ils soient portables ou à bord de véhicules, se voient de plus en plus souvent équipés de plusieurs interfaces réseau leur permettant de se connecter simultanément à plusieurs réseaux sans fil. La disponibilité de nombreux opérateurs et réseaux pour chaque technologie offre ainsi un large choix de configurations afin d'en tirer le meilleur parti. Un problème lié à la multi-connectivité émerge également, dans le sens où les flots applicatifs doivent être distribués sur les accès réseau établis. Nous appelons la combinaison de ces deux questions le *problème de gestion des flots dans un environnement multi-connecté*.

Nous avons vu dans la section précédente (et le chapitre 2) que la plupart des approches pour la sélection des réseaux d'accès sont basées sur des métriques intrinsèques aux réseaux (p. ex., force du signal ou QoS). Cependant, comme l'avance Kilkki (2008), le concept de QoE a récemment reçu un intérêt croissant. Il semble particulièrement adéquat de considérer ce critère dans le choix des réseaux d'accès et la répartition des flots afin de s'assurer

[1] Par souci de simplicité, nous utilisons le terme « métrique » même pour ces mesures pour lesquelles l'inégalité triangulaire n'est pas vraie.

que l'utilisateur du terminal en ait l'usage le plus satisfaisant possible. Suivant notre observation sur les architectures inter-couche et les décisions à la portée limitée pouvant poser des problèmes inattendus, nous considérons conjointement les deux problèmes, au sein du même modèle. Ayant ainsi la possibilité de déterminer plus spécifiquement les conditions dans lesquelles les flots applicatifs sont placés, il est aussi possible de dériver les paramètres des couches applicative et de transport et, partant, éviter les délais introduits par leurs boucles d'adaptation respectives.

Dans le chapitre 3 (page 47), nous commençons par décrire l'environnement d'un équipement mobile multi-connecté, à la fois qualitativement et quantitativement, à l'aide de données empiriques (section 3.2). Nous formulons ensuite le problème de gestion des flots dans un tel environnement et y proposons une solution prenant en compte des critères de haut-niveau observables par l'utilisateur ou directement liés à la performance des applications (section 3.3). Considérant des applications classiques telles que la navigation sur la toile ou la VoIP, pour lesquels des modèles de qualité sont connus (cf. MOS ; section 3.4) ainsi que les coûts d'accès aux réseaux et la consommation d'énergie induite, nous implantons le modèle sous forme de programmation par contraintes et réutilisons nos données empiriques pour dériver divers types de scénarios d'évaluation (section 3.5). Les résultats de l'évaluation basée sur ces scénarios, discutés en section 3.6, offrent une perspective encourageante pour l'approche proposée : en comparaison avec des techniques plus classiques telles que la sélection du meilleur réseau suivant sa QoS et un équilibrage de charge entre toutes les interfaces, notre proposition produit des solutions pour lesquelles la qualité applicative est systématiquement élevée alors que le prix et la consommation énergétiques sont maintenus à des niveaux relativement bas.

4 Protocole de transport supportant la mobilité (chapitre 4)

Suivant l'augmentation à la fois des possibilités de connectivité et la facilité d'usage des terminaux perpétuellement connectés, il y a un glissement vers l'usage d'applications réseau temps-réel telles que la diffusion multimédia en continu, la VoIP ou la vidéo-conférence. Cependant, les protocoles de transport supportant ces applications émergentes sont principalement les mêmes que ceux conçus pour des réseaux filaires statiques. Ces principaux protocoles de transport ont été présentés dans la section 2 (et le chapitre 2). Alors qu'UDP a historiquement été utilisé pour le transport de flots temps-réel, il n'est pas recommandé d'utiliser des protocoles de transport sans contrôle de congestion sur des réseaux publics tels qu'Internet (Floyd et Fall, 1999). Cependant, TCP fourni un service trop riche et inadéquat pour les flux temps-réel. En effet, sa garantie de fiabilité, retransmettant les paquets identifiés comme perdus, est offerte au prix d'un délai de transmission accru, souvent au détriment de la diffusion en continu.

Pour cette raison, DCCP (Kohler *et al.*, 2006a,b) a été proposé afin de fournir un protocole de transport non fiable mais pouvant ajuster sa vitesse d'émission au chemin emprunté. DCCP dispose d'un système permettant de sélectionner le système de contrôle de congestion. Parmi les options, TFRC (Floyd *et al.*, 2000; Widmer, 2003; Floyd *et al.*, 2008), un système reproduisant le débit de TCP suivant le modèle de Padhye *et al.* (1998), est disponible. Ce type de mécanisme basé sur une équation s'est montré bien adapté pour supporter l'envoi en continu de flux multimédia en raison de l'évolution sans heurt de la vitesse d'envoi (Floyd *et al.*, 2008). Cependant, TFRC utilise toujours les pertes de paquets comme indicateur inéquivoque de congestion. Comme mentionné précédemment, cette hypothèse est mise à mal par les réseaux sans fil et les événements de mobilité. De plus, en raison de l'adaptation graduelle du débit d'émission, TFRC ne s'adapte pas assez rapidement à des chemins réseau offrant une meilleure QoS.

Nous proposons, dans le chapitre 4 (page 75), d'étudier ce problème spécifique à la mobilité, et d'y apporter une solution. Nous commençons par une étude du comportement de TFRC lors de ces événements. Dans la section 4.2, après un bref rappel du fonctionnement interne de TFRC ainsi qu'une démonstration en simulation des problèmes en question, nous dérivons un modèle de cet algorithme. Le modèle nous permet de quantifier le nombre de paquets indûment envoyés—et irrémédiablement perdus—lors de la déconnexion, le délai avant la reprise de la transmission une fois la nouvelle connexion établie ainsi que celui nécessaire à l'adaptation à un chemin réseau offrant une capacité plus large. Nous quantifions également la capacité « gâchée », c.-à-d., qui aurait pu être utilisée si ces délais étaient nuls. Afin de diminuer ces facteurs, nous proposons, dans la section 4.3, une extension de l'algorithme de TFRC, sous forme de nouveaux états et options, permettant de suspendre l'émission des paquets avant la déconnexion du réseau courant, puis de rétablir le débit, et l'augmenter si possible, dès la reconnexion au nouveau réseau. Dans la section 4.4, nous évaluons par simulations la performance de notre extension et montrons qu'elle permet généralement de réduire à la fois le nombre de paquets perdus et la capacité gâchée après la reconnexion. Dans cette même section, nous utilisons aussi une implémentation dans le noyau Linux de notre proposition afin d'évaluer son impact sur la qualité d'une vidéo diffusée en temps-réel lors d'un cas d'usage typique. Cette démonstration met en avant le gain de qualité accessible grâce à notre extension en comparaison avec celle obtenue normalement par TFRC.

5 Précision d'une bibliothèque d'instrumentation (chapitre 5)

Ayant présenté des exemples d'éléments de décision et d'action pour notre architecture inter-couche, il nous reste à étudier la phase d'observation. L'élément en charge de cette tâche doit être capable d'obtenir les métriques représentatives des performances courantes de l'ensemble des couches de la pile protocolaire, ainsi que les informations quantitatives concernant les réseaux couramment accessibles. Ces données peuvent être utilisées immédiatement ou conservées aux côtés d'autres détails contextuels afin de supporter des algorithmes de prédictions (p. ex., Rathnayake and Ott, 2008 ou Petander, 2009).

Comme mentionné dans la section 2 (et le chapitre 2) les outils classiques permettant de mesurer activement ces métriques, ou de les observer passivement, tendent à rapporter leurs résultats dans des formats qui leurs sont spécifiques, et sont incompatibles entre eux. Ceci limite leur utilisabilité dans le cadre de notre architecture inter-couche. De plus, certains indicateurs sont déjà calculés en interne par les éléments de la pile pour lesquels ils sont pertinents. Plutôt que de re-mesurer activement ces derniers ou de les estimer, il parait plus approprié de les extraire directement des couches idoines (p. ex. le RTT observé par le protocole de transport). Certaines API permettent d'exposer ces métriques internes (comme l'instrumentation Web100 pour TCP ; voir Mathis *et al.*, 2003), mais ces interfaces sont l'exception plutôt que la règle.

Dans ce contexte, OML (White *et al.*, 2010) semble susceptible d'apporter des solutions. OML est composé d'une bibliothèque d'instrumentation et d'un système de collecte initialement développés pour centraliser les mesures provenant d'expériences réseau distribuées. Ce système fournit une API permettant au développeur d'une application, ou celui en charge d'instrumenter un outil provenant d'une tierce-partie, de rapporter n'importe quel type de donnée en utilisant le protocole d'OML. Toute information transmise de cette manière est horodatée et stockée dans une base de données, ce qui permet un accès unifié et une corrélation aisée des données provenant de plusieurs sources. Nous suggérons donc l'utilisation d'OML comme le bus de collecte des métriques observées par notre architecture inter-couche, permettant la centralisation des indicateurs de performance pertinents ainsi que des informations contextuelles appropriées et leur mise à disposition de l'algorithme de décision dans un format consolidé. Bien que décrite comme légère, la bibliothèque d'instrumentation d'OML, étant utilisée au sein même de l'application, risque d'en perturber le fonctionnement. Il est donc important de caractériser précisément l'influence de cet outil sur les applications s'en servant comme leur système de rapport.

Dans le chapitre 5 (page 103), nous développons une approche expérimentale pour comparer les performance d'outils de mesure actifs ou passifs. Nous nous en servons pour caractériser l'impact d'OML (elle-même décrite plus en détails dans la section 5.2) sur la performance des applications instrumentées, et la précision des mesures rapportées. Nous avons choisi d'étudier cet impact sur deux applications : l'outil de mesure active de capacité Iperf, et la bibliothèque d'observation passive libtrace. Nous décrivons l'instrumentation de ces outils dans la section 5.3.1. Nous évaluons l'effet de plusieurs facteurs explicatifs, comme la quantité d'information rapportée ou le débit du trafic observé (ou généré), sur les variables indicatives de la performance de l'application et de la précision de ses rapports. Les séries d'expériences que nous avons conduites pour ce faire sont décrites dans la section 5.3.2. Dans la section 5.4, nous présentons les résultats, nous assurons de leur qualité, et utilisons des analyses de variance afin d'identifier les possibles déviations des variables à expliquer en fonction des facteurs expérimentaux. Dans la section 5.5, nous discutons les domaines d'opération et scénarios dans lesquels l'impact d'OML est négligeable ou non. Pour certains cas où la fréquence de mesure ou la quantité de données à rapporter s'avèrent trop importantes pour éviter un biais, nous suggérons également quelques recommandations sur la façon appropriée d'instrumenter les applications et de mettre en place le chemin de collecte. Nos résultats supportent l'utilisation de ce système comme bus de collecte de notre architecture, et suggèrent que même une instrumentation naïve utilisant OML peut rivaliser avec un système sophistiqué de rapport développé manuellement.

6 Conclusion (chapitre 6)

Le chapitre 6 (page 127) conclut cette thèse. Nous y résumons d'abord les problèmes auxquels les équipements mobiles sont couramment confrontés. Cette thèse se place dans un contexte formé de la généralisation des accès sans fil, du nombre croissant de réseaux proposant une telle connectivité, de la portabilité de terminaux informatiques de plus en plus puissants et de la demande croissante de connectivité liée à l'émergence de nouveaux usages et applications. Les technologies, relativement anciennes, à la base d'Internet ne sont plus entièrement adéquates pour ces nouveaux environnements. Plusieurs problèmes se posent alors, tels que la sélection du réseau le plus indiqué pour supporter les besoins d'une application donnée, le manque d'adaptabilité des algorithmes de la pile protocolaire aux conditions changeantes introduites par ces décisions, et le besoin d'observer l'environnement afin de pouvoir adopter la meilleure stratégie.

Nous rappelons ensuite les contributions de cette thèse visant à aborder ces problèmes. Elles se basent sur une architecture inter-couche séparant les étapes d'observation de l'environnement, de choix des paramètres permettant une bonne performance du système, et de passage de ces paramètres aux diverses couches de la pile protocolaire. Les idées et conclusions principales des chapitres 3, 4 et 5 sont par la suite résumées :

- modélisation de l'environnement d'un terminal mobile multi-connecté, et démonstration que considérer des critères de haut niveau plutôt qu'uniquement les métriques réseau est plus adapté pour le choix des réseaux d'accès et la distribution des flots applicatifs ;
- modélisation du comportement de TFRC lors des événements de mobilité et introduction d'une extension à cet algorithme permettant une meilleure adaptation aux migrations entre réseaux hétérogènes, ainsi que son évaluation par simulation et son implémentation dans le noyau Linux de cette proposition ;
- étude de la performance d'une bibliothèque d'instrumentation démontrant que, en raison de son faible impact sur les performances des applications instrumentées et la précision des mesures rapportées, son usage est indiqué pour la phase d'observation de notre architecture.

Finalement, nous offrons des perspectives pour une continuation des travaux commencés durant cette thèse. Nous ouvrons aussi sur d'autres problèmes liés à ce thème qui sont apparus comme importants au cours du travail décrit dans cette dissertation.

Chapter 1

Introduction

1.1 Old Designs and the New Internet

The bases of the technologies in use today for the Internet date back to the early sixties, and their first implementations a decade later (Leiner *et al.*, 2009). A solid communication architecture based on separating functionalities into stacked *layers* (OSI model; Zimmermann, 1980; ISO/IEC 7498-1:1994) allowed the gradual upgrade of existing systems and the introduction of new services. At the time of the conception of the TCP/IP suite, computer networks were relatively simple and made up of a relatively small number of end-hosts. Hosts were static and had wired connections to one of their site's upstream routers through their single network adapter. In contrast, the situation today is much more complex, due to multiple factors outlined below.

Increasing Prevalence of Wireless Access The way end-hosts connect to the network is now much more heterogeneous. Wireless connectivity, both in licensed (*e.g.*, 3G or WiMAX; 3GPP TS 22.101; IEEE Std 802.16-2009) and unlicensed (*e.g.*, Wi-Fi; IEEE Std 802.11-2007) frequency bands can now be integrated within even the smallest terminals. However, the increased variability of access network properties (*e.g.*, packet loss rates or delay variability) due to various non-congestion related events such as packet collision or signal fading, has adverse consequences on the overall performance. A well known example is that of TCP which assumes that a lost packet is an indication of network congestion and reacts by reducing its packet-sending rate (Xylomenos *et al.*, 2001). This assumption was valid in a fully wired network; however, in an increasing number of situations, this is no longer the case.

Mobile Computing and Concurrent Access The reduction in size of electronic components has paved the way for highly portable yet powerful devices. Smart-phones, tablets and other portable devices provide their users with networked terminals combining both portability and computational power. Those devices change physical location as their owner, carrying them, attends to their daily tasks. Moreover, it is common for these devices to implement several wireless technologies to provide a better coverage, thus becoming *multihomed*, with a number of uplink/downlink combinations providing connectivity to the Internet, and different addresses on each interface. This dynamically changing environment creates both an opportunity to choose between different wireless networks and the requirement to take appropriate decisions in order to ensure the "Always Best Connected" (ABC; Gustafsson and Jonsson, 2003) performance and applications quality. It also puts new requirements on the ability to handle periods with lack of connectivity or transitions between different networks.

New Connectivity Modes Opportunistic communications use direct device-to-device links, rather than the infrastructure, to provide communications which are established in an *ad hoc* manner. A routing domain can be built on top of these *Mobile Ad hoc Networks* (MANET; Perkins and Royer, 1999; Jacquet *et al.*, 2002) to provide multi-hop reachability. The Internet core can also be used as a support for longer range continuation of locally initiated communications (Wakikawa *et al.*, 2007). *Delay-Tolerant Networks* (DTN; Fall, 2003) use the physical mobility patterns of the devices to relay messages to their destination over time-varying overlay networks. Recent years have also seen the emergence of *User-Provided Networks* (UPN; Sofia and Mendes, 2008), where users share their upstream connectivity by relaying packets from guest wireless clients. Depending on the resource that a user wants to access, it may prove more efficient to establish a direct connection to the topologically closest UPN rather than via the infrastructure.

Drastic Scale Increase With the lowering cost of electronics, it is now common to see several personal computers in family homes, with high speed connections to the Internet. The reduction in size of electronic components has also allowed for very small devices, such as sensors equipped with a network stack, to be deployed at will. The currently deployed IP version 4 (IPv4) only allows for 4.3 billions addresses, notwithstanding reserved ranges (Cotto and Vegoda, 2010). As of late 2011, the exhaustion of this address pool (at the RIR level) is dangerously nigh (Huston, 2011). This problem was foreseen by researchers in the early nineties, and led to the development of the next generation of the protocol (Deering and Hinden, 1998). IPv6 addresses the scale increase problem by offering an extended address space (3.4×10^{38} addresses), the correction of issues that were observed with IPv4, and a better extensibility.

Although the scale increase challenge has already been addressed with the introduction of IPv6, other issues caused by the use of wireless access and mobility are still present. These affect the TCP/IP stack, which was designed for fixed links and stationary hosts. In addition to this, the availability of concurrent access to multiple networks and new connectivity modes offer the end devices (and users) the option of more appropriately selecting the best way to establish a communication with a remote peer. Finally, new applications leveraging these evolving communications opportunities are appearing; these put additional requirements on the networks that support them. We provide an overview of such applications in the following section.

1.2 Emerging Uses

The new factors presented above allow for new and innovative uses of networked devices. This in turn introduces additional requirements for the underlying network technologies. Here we present what we perceive as the most prominent use cases.

1.2.1 Multimedia Applications

Early computer networks were used for only a handful of tasks including remote terminals, electronic mail and data file transfers. Though those applications are still relevant today, emerging multimedia applications like media streaming and *video on demand* (VoD) are becoming the norm for everyday Internet use. These are used in entertainment, education or health, to name but a few areas. Multimedia content puts an increasing strain on the network capacity, both wired and wireless, which is required to support such applications and their exponentially increasing user base.

In addition to overall capacity, a specific per-application data rate and other quality parameters like delays need to be controlled to allow for a seamless playback at the receiver. Some flexibility can be afforded by buffering data before starting playback in order to hide transient delay variations, at the cost increased average delay. However, this solution cannot be easily used in the case of interactive applications such as *voice over IP* (VoIP) or *video conferencing*, which provide two-way communications and cannot afford delays introduced by large buffers. Additionally, even with non-interactive applications like VoD, excessive use of buffering can result in potentially unacceptable (to the user) start-up delays. Other solutions therefore need be explored.

1.2.2 Intelligent Transportation Systems

A second area which has been receiving a lot of attention from the research community in recent years is that of *Intelligent Transportation Systems* (ITS; *e.g.*, Laurgeau, 2009). Enabling vehicles to communicate over standardised communication channels allows for large scale cooperation both with the infrastructure (V2I/I2V) and between vehicles (V2V). Several classes of applications are foreseen by Khaled *et al.* (2009), as outlined below.

Safety Perhaps the most important aspect, safety services could leverage the network connectivity to propagate warning messages and allow for collaborative incident management. Several European projects have already studied this aspect and proposed architectural prototypes (La Fortelle *et al.*, 2007; SeVeCom project, 2008; COMeSafety project, 2009)

Traffic and fleet management On a more day-to-day basis, vehicular networks can be beneficial to traffic monitoring and management. Services such as alternative routes, vehicle tracking or parking spot discovery can easily be envisioned. Moreover, this could pave the way to fully automated cooperative driving by allowing direct communication between closeby vehicles (Bouraoui *et al.*, 2006).

Comfort and mobility Finally, as standard technologies are deployed, legacy Internet applications can also be used. Web browsing, inter- and extra-vehicular text and voice conversations, or live video streaming from remote locations are only a few examples of future typical in-car activities.

A specific instance of MANETs, *Vehicular Ad hoc Networks* (VANET) have been widely explored (Wakikawa *et al.*, 2005; Lorchat and Uehara, 2006; Härri *et al.*, 2006; Naumov *et al.*, 2006; Ernst and La Fortelle, 2006; Mehani *et al.*, 2007) to provide communication between nearby vehicles and the roadside infrastructure. Geographic networking, where the information is routed based on the physical location of the nodes, is also getting increased consideration (GeoNet project, 2010)

The importance of having standard and interoperable technologies has been recognised in ITS (Ernst, 2006) and some of the above requirements have already been addressed by standardisation bodies. An amendment to the Wi-Fi standard, 802.11p (IEEE Std 802.11p-2010), has been ratified at IEEE to address the specificities of vehicular and *Dedicated Short-Range Communications* (DSRC). A common vehicular networking architecture is also currently being standardised at ISO as *Communications Access for Land Mobiles* (CALM; ISO 25111:2009), as well as at ETSI (ETSI EN 302 665, 2010). Most notably for our purpose, CALM specifies the use of IPv6 and its extensions for all Internet-related communications (ISO 21210:2011).

In this context, the comfort class of applications becomes a superset of the multimedia content use-case of the previous section, and shares the same issues related to both network capacity and the quality of service required to support them with an experience acceptable to users. In addition, traffic management and, to an even greater degree, safety applications introduce more stringent requirements in terms of delivery latency and reliability required from the networks used to carry their traffic. The ITS use-case therefore also requires solutions supporting increased quality of network applications in mobile contexts.

1.3 Need for Palliative Solutions to Cater for New Uses

As mentioned previously, there is a growing mismatch between the standard services that TCP/IP can provide, the current environment in which nodes implementing this stack evolve, and the requirements introduced by use-cases in mobile contexts. More specifically, we identify two main problems which need be addressed in order to bridge this gap.

Network Selection With an increasing number of available connectivity options, a choice can and has to be made on which links to establish, and where to distribute data flows so the applications' requirements can be best met according to the current network availabitily and characteristics. However these decisions are currently made irrespective of the applications needs, and are usually based on simplistic heuristics (Wasserman and Seite, 2011).

Adaptability to Changes In a mobile environment, where network links can be established and applications flows redistributed at will, the heterogeneity of the wireless technologies and network provisioning may result in wide changes in path characteristics. The current protocols of the TCP/IP stack are not designed for such sudden changes and, as they cannot adapt quickly to improvements in network characteristics, may not deliver to the applications the maximum capacity supported by the current path (Bansal *et al.*, 2001).

It is the primary goal of this dissertation to address these problems. We intend to provide elements to answer the following question.

> How to enable mobile nodes to maintain the best quality for applications and services from the available dynamically changing network resources, and support graceful degradation of the quality when the resources are scarce?

The increasingly pressing requirement to address the growing gap in the functionalities of the current Internet technologies and the needs of the emerging applications, services and connectivity modes, has led to a large body of research work in recent years. The clean slate approach, advocated by Clark (2009), amongst others, has received a lot of attention. However, the recognised problem of redesigning technology without backwards compatibility has hampered significant adoption in this direction. The second approach is more conservative, and based on incremental improvements. We perceive the latter to be a more cautious way to ensure improved functionality without a full redesign and the associated negative aspects related to the need for full deployment of new, incompatible, technologies. In the remainder of the thesis, we will propose solutions in line with this direction.

As previously noted, the TCP/IP suite is not well equipped to appropriately address all of the new requirements described in previous sections. Though it provides a functional and resilient *best effort* service, the layered approach limits the view of each component and deprives them of potentially useful information needed to provide an overall better experience. It is recognised that the inter-layer application programming interface (API) needs to be extended to expose the missing relevant information (Shakkottai *et al.*, 2003). Historically, knowledgeable users have been playing this role by adjusting the parameters of their equipment to better match their requirements. However, it is important nowadays that the devices themselves are able to adapt on behalf of non-technical users, thereby bridging the "wizard gap" (Mathis, 1999).

To this end, a large number of *cross-layer* designs have been proposed (Srivastava and Motani, 2005). Those approaches establish a specific non-standard communication channel between two or more layers of the stack, adjacent or not. However, cross-layer protocols tend to create much tighter binds than is desirable between the modified elements of the stack. Due to several unintended interactions or broken assumptions which are often created at the same time, let alone their over-specification to a subset problem, Kawadia and Kumar (2005) raised some concerns about the large scale applicability of such designs.

In this thesis, we propose to work in the same direction, but stay clear of the unintended interactions mentioned by Kawadia and Kumar (2005). Perhaps the most generic model to do so has been proposed by Thomas (2007). It makes use of the Observe, Orient, Decide, Act (OODA) loop (Figure 1.1) introduced in the military (Boyd, 1995) to represent the necessary steps in adapting to a changing environment to achieve a defined set of objectives. It consists of four different steps: the first two steps of observation and orientation include acquiring and analysing information from the environment; the next step involves making decisions based on this information and the objectives to achieve. The final step includes appropriate actions to apply those decisions.

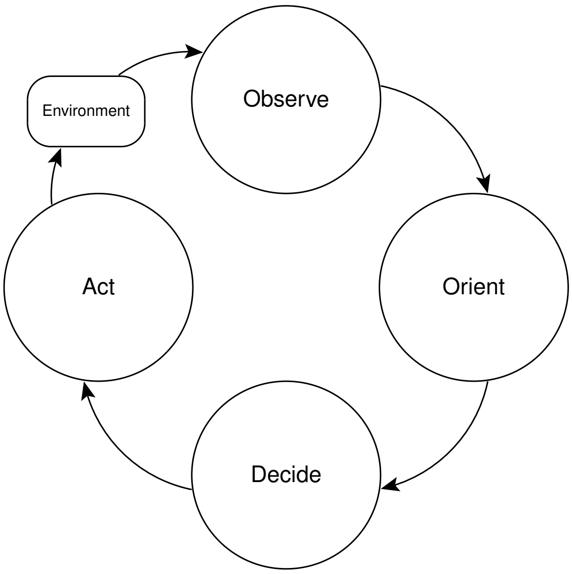


Figure 1.1 The OODA loop, describing the iterative process needed to adapt to a changing environment: Observe, Orient, Decide and Act.

1.4 Contributions: Enhancing Mobile Networks Performance

This section summarises our contributions and lists the publications relevant to each topic, while the rest of this dissertation provides details of the proposed solutions. A comprehensive list of all the publications and other contributions of this thesis is available as Appendix A (page 161).

The remainder of this dissertation is in line with the different steps of the OODA loop. We envision that a cross-layer framework such as shown in Figure 1.2 can be built from the components described in this thesis.

We first review the context and the related state of the art, as well as relevant performance metrics, in Chapter 2. The next chapters then cover the contributions of this thesis, as

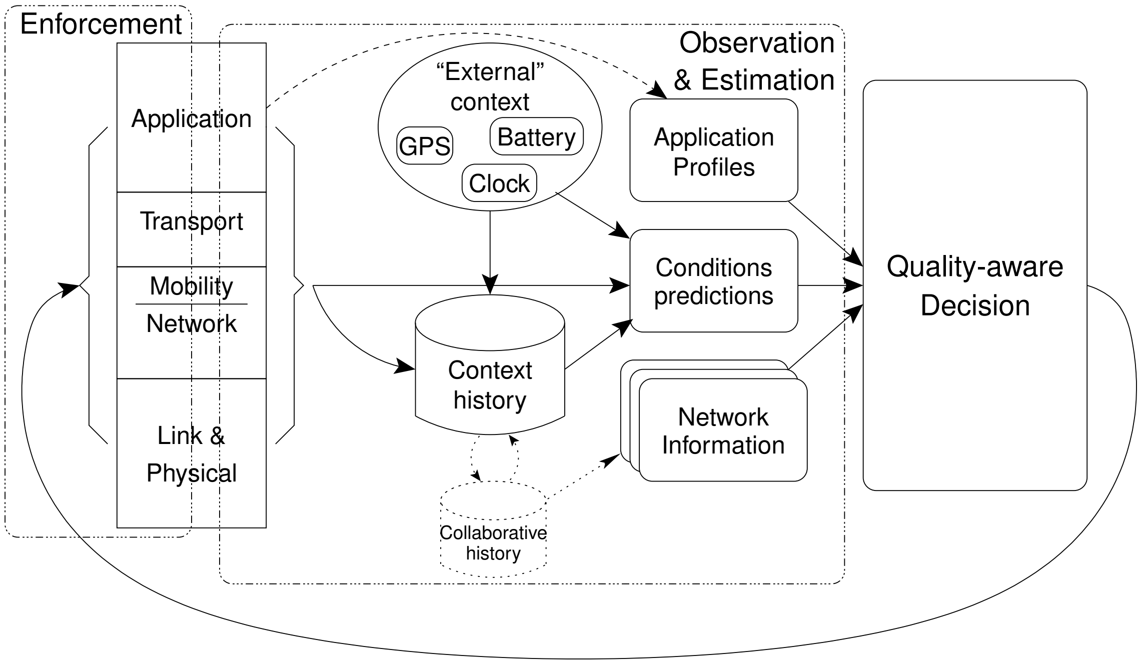


Figure 1.2 Proposed system framework linking the contributions of this thesis.

summarised below. We conclude this dissertation, and consider future research directions in Chapter 6.

1.4.1 Multi-layer Optimisation of Network Choice and Usage

Tight cross-layer designs are oblivious of the overall use of the system they are implemented in, and only focus on improving their own performance in terms of their own metrics. Only in some very specific cases do those mechanisms provide an actual improvement in the global performance from the user's point of view.

In Chapter 3, corresponding to the "decide" phase of Figure 1.1, we thus propose the use of a system external to the stack which overlooks its behaviour, and adapts the parameters at various layers in order to improve some specified performance metrics. In this thesis, we mainly focus on the *quality of experience* (QoE) a user perceives when using a mobile device.

Our first contribution is the model of the interactions between the layers of the stack, and the rest of the network. Based on this model, we conclude that the main problems which need to be addressed include the decisions on when and to which of the many available wireless networks a node should attach, how it should distribute application flows when several uplinks are available, and how application parameters and codecs should be selected to match the *quality of service* (QoS) thus made available.

The following publications are based on this contribution.

- Olivier Mehani, Roksana Boreli, and Thierry Ernst. Context-adaptive vehicular network optimization. In Marion Berbineau, Makoto Itami, and GuangJun Wen, editors, *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*, pages 186–191. IEEE Computer Society, October 2009a. ISBN 1-4244-1178-5,

- Olivier Mehani, Roksana Boreli, Michael Maher, and Thierry Ernst. User- and application-centric multihomed flow management. In Tom Pfeifer and Anura Jayasumana, editors, *LCN 2011, 36th IEEE Conference on Local Computer Networks*, pages 26–34. IEEE Computer Society, IEEE Computer Society, October 2011a.

1.4.2 Mobility-Aware Transport Protocol

Chapter 4 considers the performance of transport protocols in mobility situations. We first study the effects of network hand-offs, which create packet losses and delay variations, on the rate at which the congestion control algorithms allow packets to be sent. We then proceed to leverage the decision and control framework described above to enhance some transport layer algorithms to use the extra information and better adapt to mobility events. This corresponds to the "act" phase of Figure 1.1.

More specifically, we study the *TCP-Friendly Rate Control* (TFRC; Floyd *et al.*, 2000). This algorithm computes the sending rate using an equation modelling TCP's behaviour. Because it bases its rate computation on the same parameters as TCP, it is also sensitive to non congestion-related losses. All losses are perceived as the indication of a congestion, which results in unnecessary reduction of the packet sending rate. In addition, due to its slow responsiveness (Bansal *et al.*, 2001), TFRC has trouble adapting efficiently to better network characteristics.

As the second contribution of this thesis, we derive and validate a model of the rate evolution during a handover between two networks, in order to quantify the performance degradation. We then propose and evaluate *Freeze-TFRC*, a handover-aware congestion control mechanism. As TFRC is a rate control mechanism rather than a complete transport protocol, we focus on its implementation within the *Datagram Congestion Control Protocol* (DCCP; Kohler *et al.*, 2006a), a congestion-controlled but unreliable transport protocol.

We finally show how Freeze-DCCP/TFRC's properties make it of prime interest, in scenarios with mobility handovers, to carry real-time traffic via shared networks and preserve a good QoE.

The following publications are based on this contribution.

- Olivier Mehani and Roksana Boreli. Adapting TFRC to mobile networks with frequent disconnections. In Keith W. Ross and Leandros Tassiulas, editors, *CoNEXT 2008, 4th ACM International Conference on emerging Networking EXperiments and Technologies, Student Workshop*. ACM SIGCOMM, ACM, December 2008. ISBN 978-1-60558-210-8. doi: 10.1145/1544012.1544049,

- Olivier Mehani, Roksana Boreli, and Thierry Ernst. Analysis of TFRC in disconnected scenarios and performance improvements with Freeze-DCCP. In Jörg Ott and Kun Tan, editors, *MobiArch 2009, 4th International Workshop on Mobility in the Evolving Internet Architecture*. ACM SIGMOBILE, ACM, June 2009b. ISBN 978-1-60558-688-5/09/06,
- Olivier Mehani, Roksana Boreli, Guillaume Jourjon, and Thierry Ernst. Mobile multimedia streaming improvements with Freeze-DCCP. In Romit R. Choudhury and Henrik Lundgren, editors, *MobiCom 2010, 16th Annual International Conference on Mobile Computing and Networking, Demonstration Session*. ACM SIGMOBILE, September 2010.

1.4.3 Accuracy of a Measurement Instrumentation Library

Both previous proposals require measurements of the current network performance and environment. This corresponds to the "observe and orient" phases of Figure 1.1. To this end, we propose the use of the *OMF Measurement Library* (OML; White *et al.*, 2010).

This off-the-shelf open source instrumentation library enables direct reporting of performance metrics, with a low overhead, for corresponding instrumented applications. However, no previous work exists to evaluate the measurement overhead, or to validate the set of use-cases where OML instrumentation can provide an acceptable accuracy.

The third contribution in this thesis, presented in Chapter 5, is an experimental process to compare active and passive network measurement tools. We apply it to characterise the impact of OML instrumentations, and the quality of the reported measurements. Our results confirm that this tool is well suited for both experimental performance measurements and to obtain live feedback from a networked system, in order to collect the information necessary to make the right decisions.

This chapter is based on the work and experience described in the following publications.

- Olivier Mehani, Guillaume Jourjon, Jolyon White, Thierry Rakotoarivelo, Roksana Boreli, and Thierry Ernst. Characterisation of the effect of a measurement library on the performance of instrumented tools. Technical Report 4879, NICTA, May 2011b,
- Manabu Tsukada, Olivier Mehani, and Thierry Ernst. Simultaneous usage of NEMO and MANET for vehicular communication. In Miguel P. de Leon, editor, *TridentCom 2008, 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), March 2008. ISBN 978-963-9799-24-0,
- José Santa, Manabu Tsukada, Thierry Ernst, Olivier Mehani, and Antonio F. Gómez-Skarmeta. Assessment of VANET multi-hop routing over an experimental platform. *International Journal of Internet Protocol Technology*, 4(3):158–172, September 2009. ISSN 1743-8209. doi: 10.1504/IJIPT.2009.028655,
- Manabu Tsukada, José Santa, Olivier Mehani, Yacine Khaled, and Thierry Ernst. Design and experimental evaluation of a vehicular network based on NEMO and

MANET. *EURASIP Journal on Advances in Signal Processing*, 2010:1–18, September 2010. doi: 10.1155/2010/656407.

Chapter 2

Context and State of the Art

The work presented in this dissertation builds upon existing technologies supporting TCP/IP networks. Thus, in this chapter, we introduce this context. We first present the relevant Internet protocols and the state of the art of mobility mechanisms. We then review generic cross-layer designs, with an added focus on those supporting wireless networks and mobility, or mitigating their adverse effects. We also provide an overview of the decision mechanisms which are used in these contexts. Finally, we include a summary of performance metrics for networked mobile applications and tools for the evaluation and reporting of these metrics.

2.1 IP Networks and Mobility

This section presents the standard protocols within the TCP/IP stack designed to support end-to-end communication between Internet hosts. It then presents a summary of the work and mechanisms which have been proposed to incrementally support host and network mobility and multihoming.

2.1.1 The TCP/IP Stack

TCP/IP is an instantiation of the theoretical Open System Interconnection model (OSI; ISO/IEC 7498-1:1994), which arranges protocols in a stack according to their respective functions. Though the OSI model proposes seven layers, the TCP/IP suite is commonly seen as made of only five.

The physical layer is in charge of the medium and covers tasks such as negotiating the physical modulation or establishing electrical connections with other stations.

The data link layer controls access to the medium and ensures proper reception of messages by the station on the other end of the segment.

The network layer is where inter-networking protocols reside; this layer comprises mechanisms for node *addressing* across network segments and message *routing* between these addresses.

The transport layer maintains an *end-to-end* connection between network nodes regardless of their location in the topology; features commonly implemented at this layer include *reliability*, *congestion control* and *multiplexing*.

The application layer directly provides services to the end-user and relies on the rest of the stack for support.

It is worth noting that this layered architecture is attracting increasing criticism in light of new challenges, with numerous proposals to remodel it to current needs (*e.g.*, Ford and Iyengar, 2008; Iyengar and Ford, 2009), or replace it by taking a clean slate approach (Clark, 2009; Goldstein and Day, 2010). However, we believe it is necessary to consider the current TCP/IP stack when proposing incrementally deployable solutions.

IP as the Common Denominator

The Internet is designed for all the intelligence and control to be at the edges of the network, in the communications endpoints, while the routers in between merely relay packets towards their destination without complex processing (Saltzer *et al.*, 1984). This allows for easy incremental deployment of new technologies and applications without having to upgrade the core networks. The use of IP[1] at layer 3 has helped decouple transport layer and applications from the lower layers by abstracting their respective functionalities. This approach led to the familiar *Internet hourglass* (Figure 2.1) where IP, as the inevitable protocol, forms the waist, and a number of options are supported below and above it.

The most common version of IP at the moment, IPv4, uses 32-bit addresses which allow for almost 4.3×10^9 possibilities. However the available IPv4 address pool is nearing its end. The Internet Assigned Numbers Authority (IANA) has already depleted its address pool, and the same is expected to happen soon to the RIRs (Huston, 2011). Sharing a single public address is possible and already a reality but Ford *et al.* (2011), amongst others, recommend against it. Indeed, address sharing creates problems for applications relying on end-to-end reachability, introduces single points of failure and obfuscates the network characteristics from some control and adaptation protocols. Foreseeing this global problem, work on a new version of IP started in the early 1990s and resulted in IPv6 (Deering and Hinden, 1998). Offering a 128-bit address space (more than 3.4×10^{38} addresses), the transition to IPv6 is now unavoidable, though still rather slow (Perset, 2010).

The extended address space provided by IPv6 also enables direct addressing of *objects*, such as personal sensors (Kushalnagar *et al.*, 2007) or in-vehicle devices (Ernst, 2007). Another

[1]IP is actually a suite of several protocols, such as Internet Control Message Protocol (ICMP) or various neighbour discovery mechanisms, allowing maintenance and management of the network in addition to addressing and routing.

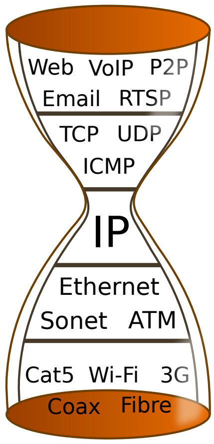


Figure 2.1 The Internet hourglass. IP has become the ubiquitous connection between heterogeneous network physical technologies and application protocols.

possibility made available, owing to this larger address pool, is the concurrent use of multiple addressing schemes. For example, geographical addressing and multicast routing is considered for information dissemination within networks of vehicles (Baldessari *et al.*, 2008; GeoNet project, 2010; ETSI TS 102 636-3, 2010). As will be seen in section 2.1.4, IPv6 also provides a much wider support for various mobility mechanisms and needs.

For these reasons, it would make little sense to study future opportunities and enabling mechanisms for IPv4. Therefore, this thesis will focus on IPv6, which is assumed whenever "IP" is mentionned without version qualifier in the rest of this dissertation.

Semantic-Rich Transport Services

The first role of a transport protocol is to provide an interface for applications to make use of the best effort service provided by IP networks. Most transport protocols also enhance this service by implementing various control mechanisms such as multiplexing, connection-oriented semantics, reliability or congestion control. This section reviews the most prominent ones, with a focus on the semantics they offer to the application and the rate control mechanisms. Table 2.1 offers a comparison viewpoint.

UDP and UDP-Lite The User Datagram Protocol (UDP; Postel, 1980) merely allows applications to carry their datagrams on top of IP packets. It however allows to multiplex several applications at a single address (*i.e.*, on a single host). UDP also provides integrity checks in the form of a checksum. However, it may not always be desirable to enforce full integrity, *e.g.*, in the case of bit-error resilient media codecs which can recover from partially altered frames. For this reason, UDP-Lite (Larzon *et al.*, 2004) has been proposed, which differs from UDP in the possibility of having only a *partial checksum coverage*.

Table 2.1 Feature sets of the common transport protocols standardised at IETF. Symbol ○ identifies optional features or extensions and ⋆ is used for alternative mandatory options.

Transport	**Connection oriented**	**Reliable**	**In order**	**Flow control**	**AIMD-like CC**	**Rate-based CC**	**Byte Stream**	**Datagrams**	**Partial checksums**	**Multiple sub-flows**
UDP								•		
UDP-Lite								•	○	
TCP	•	•	•	•	•		•			
SCTP	•	○	○	•	•			•	•	•
DCCP	•				⋆	⋆		•	•	

TCP The Transmission Control Protocol (TCP; Postel, 1981) exposes a stream interface to the application, and ensures reliable (no loss nor error) and in-order delivery of the data to the receiving end. Contrary to UDP, it is connection-oriented. This allows, amongst other features, to provide flow control, where the receiving end of the connection informs the sender of its available buffer space as a *receiver window* (`rwnd`), to prevent more data than can currently be stored to be sent.

One of the most important features of TCP is its *congestion avoidance* mechanism (Allman *et al.*, 2009). It is based on an *additive increase* of its *congestion window* (`cwnd`, the number of bytes that can be in flight) every round-trip time (RTT) and a *multiplicative decrease* when losses are observed (AIMD). It is also very commonly used as a reference point of comparison for other such mechanisms (Floyd and Fall, 1999; Bansal *et al.*, 2001; Floyd, 2008). Most notably, fairness to TCP is a very sought-after criterion for algorithms to be deployed on public networks, due to the fact that the majority of applications on the Internet currently use TCP. There is however some contention with respect to how this fairness should be assessed (Briscoe, 2007).

To adapt to a network path more efficiently without overloading it, a *slow-start* mechanism complements the AIMD mechanism. Its role is to exponentially probe the path to discover its available capacity. In this mode, TCP doubles its `cwnd` every RTT. A *slow-start threshold* (`ssthresh`) is used to detect when to terminate this phase. When `cwnd` > `ssthresh`, the congestion avoidance mechanism is used.

SCTP The Stream Control Transmission Protocol (SCTP; Stewart, 2007) has been designed to offer a more flexible choice of feature combinations than TCP or UDP. First, it can multiplex sub-streams within the same congestion-controlled (slow-start and AIMD) channel. Second, contrary to TCP, SCTP's semantic is message-oriented. It maintains the boundaries of the messages so the receiving application can use them without the need to implement its own stream delimiters. Third, in-order delivery is optional, and SCTP can be instructed to report messages to the application immediately even if earlier messages have not been received yet. Finally, though initially a fully-reliable protocol, extensions have also been introduced to support partially reliable operation (Stewart *et al.*, 2004).

DCCP The Datagram Congestion Control Protocol (DCCP; Kohler *et al.*, 2006a,b) has also been proposed to provide connection-oriented congestion-controlled datagrams streams. However, contrary to SCTP it does not enforce delivery reliability (*i.e.*, no retransmission of lost datagrams nor reordering upon arrival). This makes it a well suited transport protocol for application such as multimedia streaming, where timeliness of data arrival is more important than reliability.

DCCP has been designed with modularity in mind, and several congestion control mechanisms (identified by their *Congestion Control Identifier*, CCID) can be chosen. A TCP-like congestion control algorithm, CCID 2 (Floyd and Kohler, 2006), follows TCP's `cwnd`-based mechanism made of slow-start and AIMD. CCID 3 (Floyd *et al.*, 2006) uses TFRC (see below). CCID 4 (Floyd and Kohler, 2009) has also been proposed as a variant of CCID 3 for small packets such as used for voice over IP (VoIP).

TFRC The TCP-Friendly Rate Control (TFRC) is not a transport protocol *per se*, but an equation-based rate control mechanism (Floyd *et al.*, 2000; Widmer, 2003; Floyd *et al.*, 2008). It uses network-gathered metrics like the RTT and the number of packet losses (more precisely, the *loss event rate*) and the data rate observed by the receiver to compute the allowed sending rate, following a model equation of the throughput of TCP Reno (Allman *et al.*, 1999) under the same conditions (Padhye *et al.*, 1998).

The use of an equation-based rate control makes the rate changes smoother, which is more appropriate for the streaming of multimedia content than the abrupt changes introduced by AIMD (Floyd *et al.*, 2008). Chen *et al.* (2004) also argued that such class of rate controls tends to be more resilient to wireless losses. TFRC thus appropriately complements DCCP's unreliable datagram transport in current environments. Lochin *et al.* (2010) have also built a reliable transport protocol based on TFRC, as a smoother alternative to TCP.

Heterogeneous Physical Media

While IP was first used on fully wired infrastructures, the last decade has seen the widespread use of consumer-grade wireless devices. In addition to the common ethernet networks (IEEE Std 802.3-2008), with capacities commonly between 1 Mbps and 10 Gbps (but up to 100 Gbps), the last hop connecting user devices to the Internet can now cross a wide range of wireless technologies, in both licensed and unlicensed frequency bands. Table 2.2 summarises the characteristics of the technologies presented below.

Unlicensed Band Wi-Fi (IEEE Std 802.11-2007) provides a low cost wireless solution which can generally provide data rates from between 1 and 54 Mbps, and up to 150 Mbps

16

Table 2.2 Summary of the characteristics of common last-hop physical network technologies. Figures in parentheses are the true maximums of the standards, but are not commonly seen in user devices yet.

Technology	**Link capacity [bps]**		**Delay [ms]**
	Downstream	**Upstream**	
Theoretical (from standards)			
Ethernet (IEEE Std 802.3-2008)	1 M–10 G (up to 100 G with IEEE Std 802.3ba-2010)		(usually 1–10)
Wi-Fi (IEEE Std 802.11-2007)	1–150 M (half duplex)		—
WiMAX (IEEE Std 802.16-2009)	128 M	56 M	—
UMTS (W-CDMA; 3GPP TS 25.201)	384 k		—
3G/LTE (HSPA; 3GPP TS 25.306)	1.8–14.4 M (84.4 M)	0.73–5.76 M (23 M)	—
Measured (averages)			
IEEE 802.11	4.85 M (802.11b, no RTS; Benekos *et al.*, 2004)		15.5 (up to 20 m; Karapantelakis and Iacovidis, 2005)
IEEE 802.16	9.5 M (up to 5 km; Grøndalen *et al.*, 2007)		89–167 (Halepovic *et al.*, 2008)
W-CDMA (Prokkola *et al.*, 2009)	370 k		69–98
HSPA (Prokkola *et al.*, 2009)	2.80 M	1.30 M	40–46

with later amendments. As 802.11 uses the unlicensed[2] 2.4, 3.6 and 5 GHz radio bands, anybody can set up such a network without prior authorisation, which makes that technology highly suitable for personal or *ad hoc* communication.

Licensed Band In the licensed frequency bands, the 3$^{\text{rd}}$ Generation Partnership Project (3GPP) has specified several 3G technologies. Universal Mobile Telecommunications System (UMTS; 3GPP TS 25.201), which is already widely deployed in mobile phones, is a cell-based network system with provision to relay IP packets from the mobiles to the Internet core. 3G has been extended with new air interfaces such as High Speed Packet Access (HSPA) to support Long Term Evolution (LTE) of the network. Though the standard (3GPP TS 25.306) specifies downlink (resp. uplink) speeds of up to 84.4 Mbps (resp. 23 Mbps), most current user equipments still do not appear to support rates above 14.4 Mbps (resp. 5.76 Mbps).

WiMAX (IEEE Std 802.16-2009) is another licensed band wireless broadband technology. Though its initial specification covered fixed access provisioning only, later amendments also cater for mobile uses. Its physical medium supports downlink rates of up to 128 MBps, while little less than half of it (56 Mbps) is supported uplink. Due to licensing complexities, WiMAX has however seen little deployment worldwide, and is likely to lose the competition with the upcoming 4G/LTE from 3GPP (as already observable in data reported by Ashai *et al.*, 2011).

2.1.2 Impact of Wireless Media on TCP/IP Performance

Wireless technologies have different behaviours and characteristics as compared to wired networks. This breaks some assumptions on which the TCP/IP control mechanisms were based and causes disruptions in the performance of these protocols. The most important difference concerns the origin of packet losses but other issues arise due to the greater mobility and connectivity patterns that wireless technologies allow. This section summarises these problems and reviews some of the proposals to mitigate them.

Wireless Losses

In a well maintained wired-only network, the *only* possible source of losses is a router dropping packets due to its queue being full, that is, a congestion. In this context, packet losses can clearly be considered to be fully equivalent to congestion events.

Wireless links, however, can experience losses for other reasons such as those related to propagation impairments (*e.g.*, low signal strength or physical obstacles) or collisions at the receiver (*e.g.*, *hidden node problem*, where two senders cannot "hear" each other and transmit towards the same destination at the same time).

TCP's AIMD does not handle such losses well, as it perceives them as congestion. New update laws for `cwnd` have been proposed to replace this mechanism. TCP Vegas (Brakmo and Peterson, 1995) attempts to identify congestion by reacting to changing RTTs, un-

[2] A notable exception, 802.11p, uses the 5.9 GHz licensed band for vehicular communication purposes.

der the assumption that their increase is due to congestion-induced increasing length of queues at the intermediate routers. TCP Westwood (Mascolo *et al.*, 2001) and Westwood+ (Grieco and Mascolo, 2004) estimate the end-to-end capacity based on the rate of acknowledgements (ACKs) and adapt the `cwnd` and `ssthresh` to match this estimate. Other modifications have been proposed which extend the TCP protocol (Balakrishnan *et al.*, 1997), or address other congestion control mechanisms (*e.g.*, for TFRC, Zhang *et al.*, 2008b).[3]

Balakrishnan *et al.* (1997) note that multiple layer-2 mitigating solutions have also been proposed. Some standardised *medium access control* (MAC) protocols implement mechanisms to remediate or even avoid these losses. For example, in 802.11, each unicast frame must be ACKed by the receiving station or it will be retransmitted (a limited number of times). In addition, a signalling mechanism, *request to send/clear to send* (RTS/CTS), can be optionally enabled to reserve the channel between both nodes and mitigate the hidden node problem. Most MAC mechanisms also adjust the physical data rate depending on the channel conditions to ensure the majority of the packets can be successfully received.

These MAC techniques are however not entirely transparent to the upper layers and, if they successfully recover from a link-layer loss, it is at the price of an increased delay to transmit the packet or an overall rate reduction. Several studies have confirmed the performance degradation of TCP on these wireless media (Xylomenos *et al.*, 2001; Pilosof *et al.*, 2003; Benekos *et al.*, 2004; Franceschinis *et al.*, 2005).

Mobile Usage Patterns

Mobility, that is, the change of network point of attachment of a mobile device which has one or more active communication session, presents both problems and opportunities. We describe those in this section.

As table 2.2 has shown, the characteristics of wireless networks are very heterogeneous. In addition, MAC layers may adjust the base data rate and other devices connected to the wireless link may use a varying share of the capacity provisioned to that network. Moving active data flows from one network to another—be it using the same technology (*horizontal* handover) or a different one (*vertical* handover)—is therefore likely to abruptly change all of the parameters (such as TCP's RTT or TFRC's sending rate) that the transport protocol's congestion control mechanism has been estimating and adapting to.

As the transport protocols commonly rely on averages of the estimated parameters, there will likely be a delay in adapting to new conditions (Bansal *et al.*, 2001). The consequences of slow adaptation range from mild, *i.e.*, underutilising the new network, to serious, when the attempted rate is significantly higher than what the new network can support.

Therefore, with increased mobile usages, it becomes necessary that upper layer protocols can adapt quickly to changes of physical medium.

[3]It is somewhat misleading that some of the proposed solutions include the term "mobile" in their name while they really address *wireless* issues.

Multihoming

Mobile devices are commonly equipped with several network interfaces supporting multiple radio access technologies. Though early work focussed on selecting the best interface to activate and use depending on the context—most common user appliances still do (Wasserman and Seite, 2011)—it is also possible to activate more than one interface.

Associating each interface to a selected network of the relevant technology, the mobile device becomes *multihomed*. It is therefore reachable at several network addresses and can choose which interface to use for outgoing packets. The concept of multihoming can also be extended to networks (*i.e.*, more than one node).

Standard transport protocols like TCP cannot benefit from multihoming natively, as their notion of a connection socket is tightly bound to the network addresses in use. Standard SCTP has a mechanism to register several IP addresses for the endpoint of an association but they are only used as fail-over solutions. A similar extension for DCCP has been proposed by Kohler (2006). Modifications for TCP to support simultaneous communication over multiple paths also follow the same concept and exchange information about additional paths between endpoints (Ford *et al.*, 2010).

Identifier/Locator Problem

The emergence of mobility and multihoming highlights a problem of the use of IP addresses in transport protocols. IP addresses are *locators* indicating the topological position of a node within the network. However, rather than with a specific location, transport-layer connections are established with a given host. In a purely static topology where each host has a single unchanging link to the rest of the network, as the Internet used to be, it is safe to also use this IP address as an *identifier* for the endpoint (Zhang *et al.*, 2007).

However, now that a single host can change locators during the lifetime of some connections (mobility) or be reachable via several of them (multihoming), the problem of unambiguously identifying the node arises (Lear and Droms, 2003). Moreover, as transport protocols were designed to directly manipulate locators as identifiers, establishing and using multihomed associations requires heavy modifications to the current protocol stack implementations.

2.1.3 Mobility and Multihoming Support

Many mobility mechanisms have been proposed to mitigate the above-presented addressing problem. Such proposals can be found at almost all layers from network to application of the OSI model (Nazir and Seneviratne, 2007). Even though there are various arguments in favour of mobility and multihoming support residing in a specific layer stack (Eddy, 2004b), the question remains largely open.

Mobility at Various Layers

Network layer mobility solutions allow nodes to maintain transport layer connections by using some network addresses as identifiers (Akyildiz *et al.*, 2004; Perera *et al.*, 2004). Two classes of network layer solutions can be identified. Some solutions create overlay networks to carry packets from the topological location of an identifier (the address when the communication was initiated) to a locator (the current address). Common examples include *Mobile IP* (MIP and MIPv6; Perkins, 2002; Johnson *et al.*, 2011) or *Mobility Support using Multicasting in IP* (MSM-IP; Mysore and Bharghavan, 1997). The second class introduces new database elements to map identifiers to their current locators and answer queries from correspondent nodes or routers along the way. Proposals in this class include *Mobile IP with Location Registers* (MIP-LR; Jain *et al.*, 2001), *Location Independent Networking for IPv6* (LIN6; Teraoka *et al.*, 2003) and the *Locator/ID Separation Protocol* (LISP; Farinacci *et al.*, 2011a,b).

In order to yield better performance and reduce the global network traffic caused by mobility, various *IP micro-mobility* schemes have also been proposed (Reinbold and Bonaventure, 2003; Akyildiz *et al.*, 2004). They follow similar models as for global mobility and hide a device's movement within an administrative domain. *Mobile IP with Regional Registration* (Gustafsson *et al.*, 2004), *Hierarchical Mobile IPv6* (Soliman *et al.*, 2005) and *Intra-Domain Mobility Management Protocol* (IDMP; Misra *et al.*, 2000, 2002) extend the overlay network approach of MIP, while *Cellular IP* (Valkó, 1999; Campbell *et al.*, 2000) and *Handoff-Aware Wireless Access Internet Infrastructure* (HAWAII; Ramjee *et al.*, 2000, 2002) maintain local collaborative databases in the form of the domain's routers' routing tables. *Proxy Mobile IPv6* (Gundavelli *et al.*, 2008), is a hybrid solution, leveraging MIP messages, but letting the visited network maintain associations on behalf of the visiting node. It is recognised as well suited for operator-centric mobility management with light terminals and has been proposed for use in 3GPP architectures (3GPP TS 29.275)

There are two main types of transport-layer solutions to support mobility. One type extends the protocols messages with secrets to allow re-establishment of an existing session from another IP address. The TCP Migrate options (Snoeren and Balakrishnan, 2000), TCP-R (Funato *et al.*, 1997) and TCP Mobility (Eddy, 2004a) are examples of such mechanisms. Multihoming extensions to this concept propose to simultaneously register several endpoint addresses with the peers for TCP (Matsumoto *et al.*, 2003), SCTP (Riegel and Tuexen, 2007; Budzisz *et al.*, 2008) or DCCP (Kohler, 2006). Another approach consists in splitting the connection at a proxy element. I-TCP (Bakre and Badrinath, 1995), M-TCP (Brown and Singh, 1997) and MSOCKS (Maltz and Bhagwat, 1998) are based on this approach. Despite its name, *Session Layer Mobility* (SLM; Landfeldt *et al.*, 1999) also fits in this category.

At the application layer, proposals to support mobility maintain state information across mobility-induced losses of connection, and manage correspondent peers with multiple network addresses by introducing application-layer identifiers for the sessions. The *Session Initiation Protocol* (SIP; Rosenberg *et al.*, 2002; Sparks, 2003) is a generic session management protocol, which has been extended by Schulzrinne and Wedlund (2000) to provide *Application-Layer Mobility*. Tourrilhes (2004) introduces an application-aware proxy service able to re-establish application sessions from a mobile device to the infrastructure and resend the data which may have been lost due to the change of network.

Finally, some proposals argue for the specific introduction of a new layer to support naming or mobility (Henderson, 2003). The *Host Identity Protocol* (HIP; Moskowitz *et al.*, 2008) introduces a layer based on cryptographic identifiers to decouple transport bindings from network addresses. Shim6 (Nordmark and Bagnulo, 2005; Launois and Bagnulo, 2006) introduces a *shim*[4] layer which is located between the network and transport layers, and performs the necessary lookups to find the currently valid locator addresses for a given identifier. Shim6 however only addresses the multihoming problem.

2.1.4 Which Layer Is the Most Appropriate for Incremental Mobility?

Recalling the hourglass shape (Figure 2.1 on page 13) of the IP stack, it is clear that the network layer is the only one for which a single common protocol exists. It is also the layer for which locators are meaningful. Although the upper layers also deal with addresses in the same scope, they misuse them as identifiers. We note that the lower layers use their own specific addressing schemes, however these are not visible to higher layers.

To provide incrementally deployable solutions, introducing mobility mechanisms at, or around, the network layer seems to be the wisest approach. Furthermore, providing IP-like identifiers to the upper layers and transparently mapping them to relevant locators would avoid the need for the large number of applications and transport protocols to be modified in order to become able to support moving hosts. With appropriate support, simple multihoming can also be provided by associating more than one locator to an identifier.

In this respect Mobile IPv6 (MIPv6) is the most promising solution. Its featureful set of extensions further supports this statement. We therefore choose this solution as the basis for our work, and present its operation, terminology and extensions in more detail in the next section.

MIPv6 and Its Extensions

Mobile IPv6 allows to establish bi-directional IP tunnels between a *mobile node* (MN) and its *home agent* (HA). The MN has a particular IP address, the *home address* (HoA) which it uses as an identifier. The HA thus intercepts any packet for the MN's HoA, and encapsulates it in the tunnel towards the *care-of address* (CoA) of the MN, its current locator, as negotiated during the *binding update* (BU) phase.

This mobility scheme only requires the addition of HAs in the infrastructure. Performance can however be improved if the *correspondent node* (CN) also supports MIPv6. In this situation, *route optimisation* (RO; Arkko *et al.*, 2007) can take place to avoid the *triangular routing* problem, where all packets to the MN have to go through the HA. The top and left parts of Figure 2.2 graphically provide an overview of these concepts.

[4] A shim is a piece of code which emulates an application programming interface (API) used by the layers above it and either modifies or translates the parameters it is passed. Shims are usually used to transparently hide API compatibility issues or implement new functionalities on behalf of the upper layers.

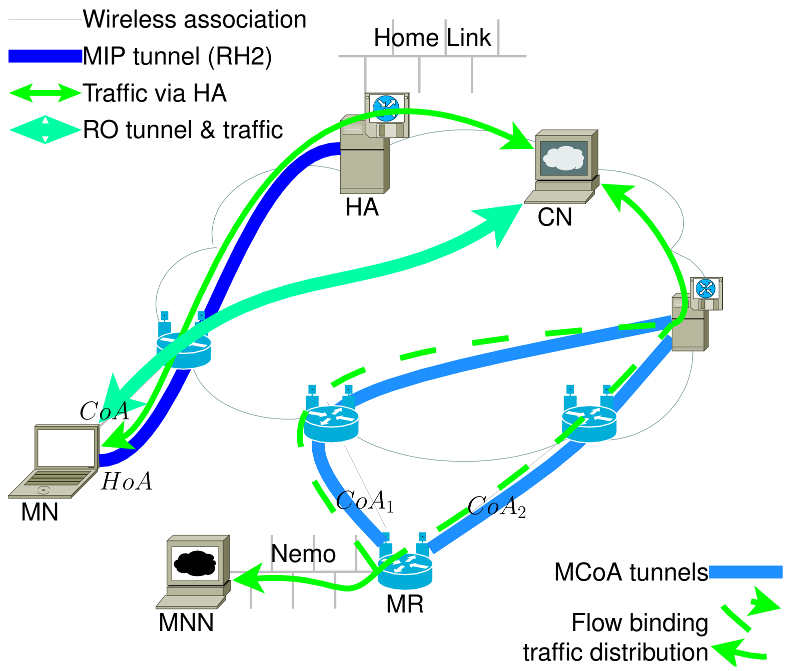


Figure 2.2 Mobile IPv6 and NEMO concepts. A mobile node (MN) can communicate in two ways with a correspondent node (CN). A tunnel can be established with its home agent (HA) in which the latter forwards all packets addressed to the MN's home address (HoA) to its care-of address (CoA). If it supports it, the CN can be the endpoint of a route optimisation (RO) tunnel to avoid the triangular routing problem. With NEMO, a router can become mobile (MR), and transparently forward traffic on behalf of its mobile network nodes (MNN). Multiple care-of address registrations and flow binding allow either MN or MR to be connected to several networks and finely control the distribution of the traffic on those uplinks.

MIP is completely transparent for the higher layers as long as they only manipulate HoAs. However, it introduces delays during handovers, while the CoA changes (Lee *et al.*, 2004). MIPv6 Neighbourhood Routing for Fast Handoff (Yegin *et al.*, 2000) and Fast Handovers for Mobile IPv6 (F-MIP; McCann, 2005; Koodli, 2008) have therefore been proposed to reduce this delay. In these schemes, an MN, once it has established connectivity, keeps sensing the medium and pre-emptively negotiates associations with other networks' *access router* (AR).

Multiple extensions to MIPv6 have been proposed. The most important is the extension to *IPv6 NEtwork MObility* (NEMO; Devarapalli *et al.*, 2005). Following the principles of MIPv6, a *mobile router* (MR) maintains a binding with its HA. Traffic to all its guest *mobile network nodes* (MNN) is then similarly forwarded via the MR's CoA.

The registration of *Multiple Care-of Addresses* (MCoA; Wakikawa *et al.*, 2009) brings multihoming to mobile nodes and networks as transparently as MIPv6 did mobility. *Flow Binding* rules (Tsirtsis *et al.*, 2011; Larsson *et al.*, 2009) allow an MN (or MR) to negotiate with its HA and enabled CNs to finely control the distribution of both upstream and downstream flows. The bottom-left part of Figure 2.2 shows the NEMO and MCoA concepts. The latter can however also be used in the context of a single MIPv6 MN.

Mitsuya *et al.* (2007) present a compelling example of an architectural framework which enables exploiting the full benefit of this type of approach.

Manner and Kojo (2004) and Ernst and Lach (2007) have defined a terminology for elements and events involved in MIPv6 and, respectively, NEMO. In the rest of this dissertation, we use these terms as defined in these documents.

2.2 Cross-Layer Designs to Improve Performance of Mobile Networks

Though it solves the semantic identifier/locator problem, introducing mobility at the network layer does not address performance issues that often occur when changing access networks. While only a small number of changes, *e.g.*, congestion level, may occur over time in fixed networks, a broader range of characteristics may vary in wireless and mobile networks, based on the environment or other conditions.

As it is currently specified and implemented, the API between layers of the TCP/IP stack, historically designed for wired links, does not allow for the relevant information about these changes to be shared across different layers. It would however be beneficial to the performance of applications and services if this information were available. To address this issue, a class of solutions based on cross-layer interactions has been the focus of an extensive research effort in the last decade (Shakkottai *et al.*, 2003, *e.g.*,).

Cross-layer designs extend the amount of information two or more layer implementations can exchange in order to provide more insight about their current conditions and allow for a better adaptability. Based on the work of Srivastava and Motani (2005), cross-layer approaches can be classified in four main classes: exchange of information between layers (upwards, downwards, or in both direction), merging of adjacent layers, coupled design of protocols and vertical calibration (Figure 2.3). They also identified three implementation approaches: extension—often in a very *ad hoc* way—of the API, addition of a shared database-based vertical plane or complete reorganisation (Figure 2.4).

The remainder of this section summarises the research work on cross-layer designs and is organised following the classification shown in Figure 2.4.

2.2.1 Direct Communication Between Layers

In this section, we review direct communication between layers as well as approaches which implicitly rely on assumptions from other layers. These two classes are by far the most common. This section is organised according to the layer which is the primary focus of the introduced enhancement.

Radio Link

Radio links are instantiated by the two lower—physical and link—layers. The typical 802.11 link layer selects the channel modulation and transmission rate based on the success rate of recent packet (re-)transmissions (Kamerman and Monteban, 1997). Rather than using this purely MAC-centric approach, Holland *et al.* (2001) propose to pass meas-

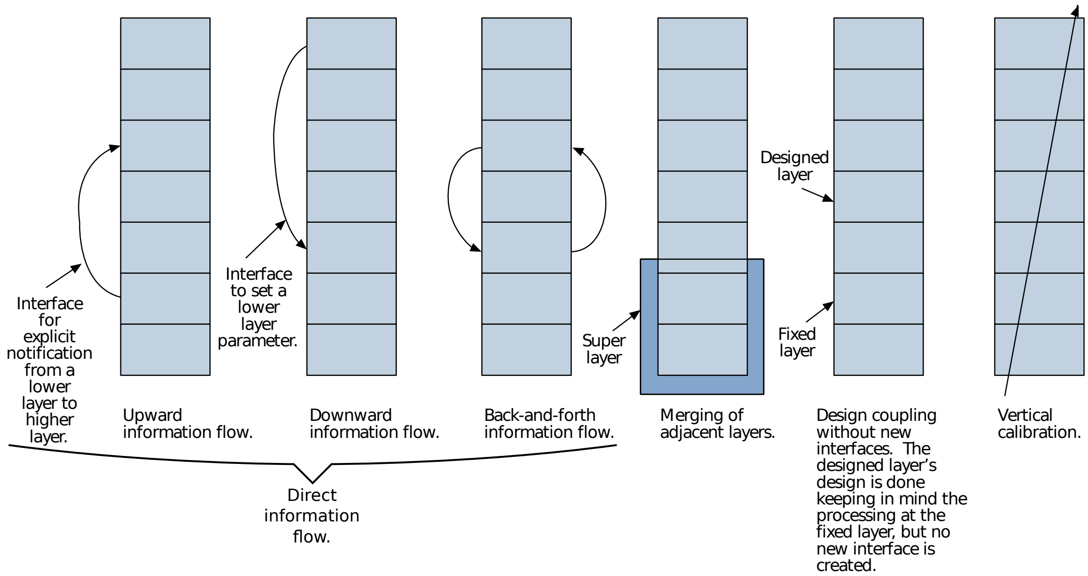


Figure 2.3 Four main classes of cross-layer designs: direct information flow (upwards, downwards, or in both direction), merging of adjacent layers, coupled design of protocols and vertical calibration (adapted with permission from Srivastava and Motani, 2005).

urements of the physical characteristics of the medium upwards for use at the MAC layer. Their *Receiver-Based Auto-Rate* (RBAR) extends the 802.11 RTS/CTS MAC messages for the receiver to measure the *signal-to-noise ratio* (SNR) and inform the sender to adjust its sending rate accordingly. Sadeghi *et al.* (2005) introduce an *Opportunistic Auto-Rate* (OAR) to further extend the proposal and allow stations to equally share the medium by controlling the time they hold it. Pairs with higher SNRs transmit at a higher rate and can therefore send more than one back-to-back packets in the same time it would take for one packet on a worse link. While the former proposal improves transmissions over bad quality channels at the cost of more signalling, the latter leverages the same approach to reduce the per-packet overhead when the channel is good, and fairly shares the spectrum. Camp and Knightly (2008) propose an experimental framework for the implementation and evaluation of such cross-layer rate-adaptation mechanisms such as those above. They pay particular attention to vehicular networks and show that SNR-based rate adaptation is indeed better suited than loss-based in mobile environments.

ElBatt *et al.* (2000) present a downward information exchange scheme to control the physical transmission power based on the achieved throughput between nodes in wireless *ad hoc* networks. The goal is to maintain connectivity while limiting the transmit power in order to avoid interference to by-standing stations, as well as the battery drain. This enhanced MAC protocol enables each node to periodically select a set of closest neighbours and inform them on a control channel. The selected neighbours then adjust their transmission power to achieve an acceptable *signal-to-interference ratio* (SINR) at that receiver. ElBatt and Ephremides (2004) further extend this concept by complementing it with a finer-grained power control with channel access scheduling to maximise the number of non-interfering communications within each time slot.

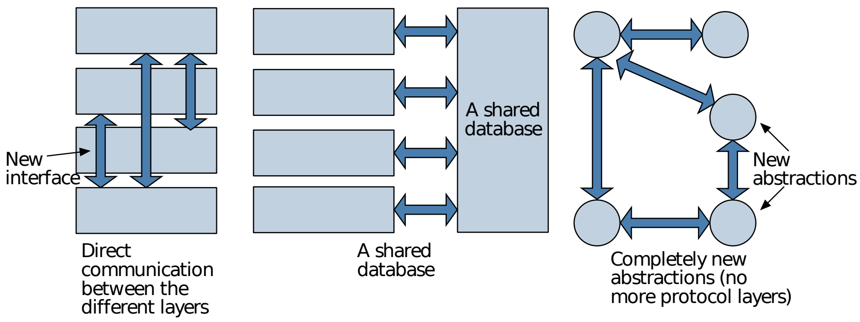


Figure 2.4 Three approaches for cross-layer implementation (reprinted with permission from Srivastava and Motani, 2005).

Network Layer and Routing

Routing protocols generally maintain a table of neighbours which can potentially forward traffic, and their associated metrics. While the metric for static wired networks is most commonly the number of hops to reach a target network, mobile networks comprise links of variable quality which cannot be accurately represented by a static metric. New metrics based on cross-layer information about the radio link have therefore been introduced to better support Mobile *Ad hoc* Networks (MANETs) and Vehicular *Ad hoc* Networks (VANETs) (Qin and Kunz, 2004).

De Couto *et al.* (2003) introduced the Expected Transmission count (ETX) metric. ETX is a compound metric incorporating the forward and reverse delivery success to estimate the expected number of retransmissions for a packet to be successfully received and acknowledged by the destination. The delivery ratios are measured by link-layer probes and passed upwards to the routing protocol. A similar metric has been introduced by Draves *et al.* (2004): the Expected Transmission Time (ETT). ETT is an adjustment of ETX which also incorporates the raw data rate of the radio links. Therefore, rather than the number of retransmissions, it estimates the time for a packet to be successfully transferred, in order to be able to account for links of different speeds, and gives preference to routes with a higher delivery speed. Both ETX and ETT are shown to efficiently replace more standard metrics in multi-hop *ad hoc* networks.

The work of ElBatt *et al.* (2000) uses the negotiated transmission power as the metric at the network layer to provide a minimal-power routing.

Zhou *et al.* (2005) propose a generic *cross-layer route discovery framework* for MANETs. It is a reactive protocol, that is, it tries to find a route to a destination only when traffic towards this destination exists. However, rather than relying on a unique route metric, the possible relays evaluate their utility based on information or requirements from upper layers (*e.g.*, capacity or delay for the application, throughput of the transport or local node's battery level).

Vehicular network nodes equipped with sensors can use VANETs to exchange their readings. However, the short validity time of some of these readings requires a quick delivery for which Wiegel *et al.* (2007) suggest a tighter link between MAC and routing protocols.

Their *cross layered cluster based packet forwarding* proposal exposes neighbours detected at the link layer to the route discovery mechanisms to allow the creation of clusters of vehicles.

In another attempt to provide delay-based quality of service (QoS) support to MANETs, Tian *et al.* (2005) introduce a cross-layer approach to QoS provisioning by increasing the interactions from the application, down to the link via the network layer. At the application layer, the flows are classified depending on their delay requirements and the current performance of the network. The network layer protocol then marks the packets accordingly. This mark is used by a MAC-layer scheduling algorithm, both locally and when forwarded. If the deadline for a packet to be delivered is missed, it is also pro-actively dropped. This approach shares similarities with the classification mechanism of 802.11e (IEEE Std 802.11-2007, sec. 5.4.5) but allows intermediate routers to also schedule the packets appropriately.

With a similar objective in energy-constrained MANETs, Hortos (2003) derives a more generic bidirectional cross-layer optimisation of routing policies to better control the QoS given to each class by focusing on the dependence of protocol performance on lower layer metrics. Likewise, Johansson and Xiao (2006) present a joint optimisation of the achievable end-to-end rate and power consumption in multi-user wireless networks. Using mathematical models of the network layer and radio link, they propose a non-linear technique which explores the parameter space in terms of flow routing and packet scheduling which, they show, finds a globally optimal and fair solution.

Mobility

In addition to the signalling needed to accomplish the handover described in Section 2.1.3 (page 19), mobility presents other challenges. One common aspect of mobility situations is the need to identify when to start and how to drive a handover. Deciding when to hand off a network and appropriately choose the next network to use is a complex problem for which we review the decision mechanisms later, in Section 2.3.1 (page 32). However, many of the proposed techniques rely on information acquired from other layers, which we detail here.

Perhaps the most common approach is based on obtaining *received signal strength* (RSS) information from the radio-link layer which is then used by the upper network/mobility mechanism to decide whether to prepare for a hand-off. A number of proposals use this cross-layer information as the basis for all or part of the decision to change the currently attached network (Mäkelä *et al.*, 2000; Ylianttila *et al.*, 2001; Gwon *et al.*, 2002; Park *et al.*, 2003; Mohanty and Akyildiz, 2006).

In an attempt to further improve the seamless nature of F-MIPv6-based handovers, Montavont and Noël (2006) and Krishnan *et al.* (2007) have proposed to use layer-2 call-backs for the mobility system to be aware in real-time of the status of the link without relying on periodic layer-3 messages. This work has then been oriented towards an abstraction of layer-2 states and events (Gogo *et al.*, 2006; Teraoka *et al.*, 2008) to allow for less dependence of the mobility subsystem on the type of link layer. With this level of information, layer-3 mobility algorithms can prepare the handover with other ARs in advance and feed instructions back to the layer-2 to hand off at the most appropriate time (*i.e.*, before the signal strength has been reduced to too low a level to maintain the link). Puttonen *et al.* (2005) and Sooriyabandara *et al.* (2008) also propose generic link layer abstractions which can be used for handover purposes.

Transport Layer

A large body of cross-layer designs exist to adapt various transport protocols to wireless networks (Sarolahti *et al.*, 2007).

Explicit Congestion Notification (ECN; Ramakrishnan *et al.*, 2001) may be the best known example of collaboration between the network and transport layers. It relies on the IP routers' marking the packets when their queue approaches congestion. The TCP receiver reports this information to the sender which uses it in its congestion control to reduce the rate before a loss occurs. ECN has been used as the basis of multiple proposals to differentiate wireless losses from congestion-induced ones, with mixed results (Biaz and Wang, 2004; Ramadan *et al.*, 2009).

To alleviate the impact of serious degradations of the wireless channel on the TCP congestion control, Goff *et al.* (2000) propose to detect such events at the receiver and temporarily suspend ("freeze") the sender. To this end, they leverage the window-based flow control mechanism of TCP by advertising a null `rwnd` (*zero-window advertisement*) when the wireless channel fades away. When the channel is restored to a usable level, the receiver sends a non zero-window advertisement, therefore resuming the sender's operation with the same congestion window. Baig *et al.* (2006) extend on this work for vehicular networks and analytically show that, with appropriate disruption prediction, Freeze-TCP provides a real performance gain over a large range of disconnected periods. Park *et al.* (2009) also extend this work in the case of vertical handovers. They point out that the achievable capacity on the new network may be notably different than the previous one, and propose to adjust the congestion window based on the difference in the measured RTT, in a way inspired by TCP Vegas.

Zhang *et al.* (2008a,b) show that, as for TCP, losses due to contention in wireless LANs disrupt TFRC's rate estimation. They identify the specific case where TFRC's natural rate increase during loss-less periods leads to the wireless medium being saturated. The thus created losses lead TFRC to reduce its rate, and eventually results in an oscillating behaviour. The authors therefore suggest to limit the transport protocol's sending rate control law to the data rate currently achievable by the underlying wireless link. They further extend the proposal by adding a similar constraint in order to fairly share the wireless link with other users. Lochin *et al.* (2006) also extend TFRC with knowledge about the user's QoS agreement so the rate control always uses the minimum throughput guaranteed by a DiffServ Assured Service classification. The algorithm controlling the rate is adapted to take into account this guarantee, using it as a minimum.

Using the upwards triggers proposed by Teraoka *et al.* (2008), Han and Teraoka (2008; 2009) propose to inform the SCTP sender about upcoming disconnections, without relying on timeouts, so it can preemptively switch its communication to the backup path. The proposal is then extended to configure fall-back addresses for each wireless network in range, thus allowing for very short hand-off times.

Chiang (2004) introduces a distributed power-management scheme which identifies and eliminates bottlenecks impacting TCP throughput in MANETs. Based on the assumptions of the TCP Vegas algorithm, each node evaluates the evolution of their current queueing delay and transmits this information to the network. Using an aggregate of such information from all nodes, every node adjusts their transmit power to favour the neighbours with the largest queue. That node can subsequently increase its power by the largest amount to benefit from a larger data rate, and flush its queue faster.

Applications

The most common proposals using cross-layer approaches for applications have been for multimedia content, due to the close link between the network capacity and delays and the quality of such applications. Models of the radio link layer and losses at the different supported rates have been used in a number of cross-layer proposals for applications with a video component over wireless links. They mostly follow the up-/downwards approaches in Srivastava and Motani (2005)'s first main class, but also show interesting examples of design coupling and vertical calibration.

Some proposals adapt the application parameters to the characteristic of the lower layers. Van der Schaar *et al.* (2003) model the video codec and adapt its parameters in terms of redundancy (using *forward error correction*, FEC), frame size and packet retransmission to maintain an end-to-end video transmission over wireless links based on their current characteristics. Kofler *et al.* (2008) take a similar approach but use abstracted information and metadata containers standardised as part of the MPEG-21 multimedia framework (Burnett *et al.*, 2006) for end-to-end signalling. They also use TFRC at the transport protocol, due to its non bursty profile (Bansal *et al.*, 2001), for which the knowledge of the rate control law allows to determine an estimate of the current transmission rate—an example of Srivastava and Motani (2005)'s coupled design, in addition to upward information exchange. TFRC is used in a similar but more direct fashion (no coupled design) by Sarwar *et al.* (2011) in a DCCP-based cross-layer tool which matches the encoding rate of an H.264 video to the current sending rate of the transport protocol.

Conversely, other approaches adapt the lower layers' parameters based on information on the application's requirements. Khan *et al.* (2006) take into account knowledge from the application and its codec to adapt the link-layer rate for each group of pictures, in order to trade speed off for reliability depending on the importance of the frames. Loiacono *et al.* (2010) propose a similar method which adapts the physical rate depending on the observed quality of the link in terms of SNR and *bit error rate* (BER). Finally, Saesue *et al.* (2010) add the 802.11e (IEEE Std 802.11-2007, sec. 5.4.5) access class, to which each group of pictures is allocated, as an adjustable parameter to enhance video transmission quality.

Solutions which attempt to more finely control both the applications' and lower layers' parameters at the same time have also been proposed. Ci *et al.* (2008) introduce an analytical model of multimedia communication over wireless networks. They use the insight gained from that model to derive a cross-layer mechanism for video transmission over MANETs. They propose to adjust the video encoding bit-rate, select the network path with the highest SNR, and minimise the radio-link delay along that path. Considering a wireless network used by multiple users to receive on-demand video streams, Özçelebi *et al.* (2007) propose a cross-layer packet scheduling approach to maximise the overall users' satisfaction. The solution is in charge of deciding which user flow gets transmitted on the radio link for each time-slot and selects the encoding size for the transmitted groups of pictures in order to maximise the video quality and avoid empty buffers leading to unwanted pauses in the playback.

2.2.2 Vertical Plane Approaches

Overall, the approaches overviewed in the preceding sections are very specific in the problem they attempt to solve and the way they are suggested to be implemented does not allow for much flexibility. In contrast, other works, which we review in this section, have taken a more generic approach, which in most cases can be mapped to Srivastava and Motani (2005)'s shared database architecture.

To address the problem of long-term stability of the Internet in terms of congestion, Balakrishnan *et al.* (1999) introduce an architecture based on a *Congestion Manager*. It relieves the transport layer from having to do congestion control locally for each connection by implementing a global information system about the achievable rates on the paths in use. It also provides similar notifications to applications, thereby removing the need for *ad hoc* adaptation loops. More generally, noting the poor applicability of the default TCP parameters in some scenarios, Tierney *et al.* (2001) and Dunigan *et al.* (2002) propose the use of a TCP tuning daemon. Based on the samples from a network monitoring framework (*e.g.*, delays or capacity) and an instrumented TCP (Mathis *et al.*, 2003), this daemon adjusts parameters of the TCP instances, such as buffer sizes or number of streams, to best fit the characteristics of the network path they use.

Within the MobileMAN project (Conti *et al.*, 2003), which explores the potential of metropolitan MANETs, MobileMAN project (2004) and Conti *et al.* (2004) design a cross-layer architecture based on a sideways network status information bus which the legacy protocols of the stack can be extended to interface to. In a more generic context, Chang and Gaydadjiev (2008) propose a similar vertical bus. Both proposals therefore allow legacy protocols to obtain more information relevant to their operation about the overall network state without changing their interfaces to the rest of the stack.

Contact Networking, proposed by Carter *et al.* (2003), is intended to support proximity mobility by providing agility in the wireless connectivity it uses, as a complement to MIP for local connections (*i.e.*, not involving the core Internet architecture). Within a connectivity manager spanning the link and network layers, it maintains a database of neighbours and the technologies they support. This information allows the mobile nodes to establish new radio connections and configure the network layer appropriately, as needed to maintain communication with a peer when the current link breaks.

With a focus on NEMO-based mobility in vehicular networks, Ben Rayana and Bonnin (2008) introduce the concept of *Mobility Aware Applications* (MAwA). This awareness is realised by introducing a management component spanning both the MR and its MNNs. This new component lets the MNN-hosted applications state their requirements and, based on observations of the current conditions, it then calls decision modules and configures accordingly the link and network layers as well as the mobility infrastructure. MAwAs can also receive information on the current conditions for use in their own adaptation loop.

Recently, approaches based on such vertical layers have seen an increasing interest from standardisation bodies. IEEE specified 802.21, primarily to support media-independent handovers (MIH; IEEE Std 802.21-2008; Piri and Pentikousis, 2009a). Though the initial objective was to leverage ideas such as link layer triggers (Teraoka *et al.*, 2008) and preemptive network configuration (Koodli, 2008), 802.21 provides a framework for information passing and remote function execution which seems well suited to more generic cross-layer interactions. Piri and Pentikousis (2009b) have introduced a discussion on implementation considerations for some platforms and projects such as OpenMIH (Lopez and Robert, 2010) or ODTONE (Corujo *et al.*, 2011) already provide working components of the standard for common operating systems.

More specifically for vehicular communication, the ISO Communications Access for Land Mobiles (CALM) ITS-Station reference architecture includes a similar management plane (ISO/CD 24102:2008; ISO 21217:2010) providing information to support cross-layer configuration of radio and network interfaces in order to maintain connectivity between on-board applications and the infrastructure over mobility events, amongst other things. Purposefully, both IEEE 802.21 and ISO CALM specify only APIs and the way information is exchanged, without any provision on how any decision is made.

2.2.3 New Abstractions

For the "new abstractions" in Figure 2.4, Srivastava and Motani (2005) consider all deviations from the current TCP/IP stack to be some form of cross-layer design. In line with them, we review notable architectures which significantly diverge from the TCP/IP stack model. We note that some proposals keep the wireline TCP/IP message structure, with headers and encapsulation, and only change a node's internal processing, while others also depart from this data serialisation.

Hutchinson and Peterson (1991) argue for a much finer-grained protocol selection. Their proposed x-kernel framework allows the developer to select a set of features to support each flow, therefore building a protocol stack dedicated to each communication. A more recent proposal by Ford and Iyengar (2009) introduces a similar framework where the communication peers can negotiate which of the protocols implemented on both ends are to be used for each application flow. In earlier work, Ford (2007) has also proposed to decouple the function of congestion control from the semantics of the application flows by providing a connected channel sub-layer on top of which multiple types of stream- and datagram-based exchanges can be transported.

Moors (1998) proposes a decomposition of the protocol stack into functional modules to avoid code and semantic elements (*e.g.*, transport- and application-layer acknowledgements) duplication. Similarly, Braden *et al.* (2003) introduce a role-based architecture where all the communication protocols can access all information (headers) of a message to decide how to best handle it. Such an approach is used in the Click router architecture (Kohler *et al.*, 2000) and shown to manipulate IP packets more efficiently. In his PhD thesis, Grunenberger (2008, chap. 6, sec. 2) introduces a generic packet manipulation solution, PACMAP. Each packet received from the radio link layer is extracted from the interface's buffer, including headers down to the MAC layer. These can then be inspected and used by several functions implementing different protocols. We note that PACMAP could be a good candidate to implement the two former proposals and any type of message format.

Mitola and Maguire (1999) have first introduced the concept of cognitive radios using an abstract representation of their environment to adjust their parameters. In his DTech thesis, Mitola (2000) presents a framework allowing radios to sense their environment and negotiate with their peers to select the most appropriate communication parameters at the current time. Perhaps the most interesting possibility of this approach is the opportunity of using the full frequency spectrum, including reserved bands when they are found to be free. Cognitive networks, first introduced by Thomas *et al.* (2005), are a generalisation of this concept to the rest of the stack to efficiently support the end-to-end communication goals. In contrast with more common cross-layer approaches, cognitive networks need to account for multiple conflicting optimisation criteria (Thomas *et al.*, 2006).

Unfortunately, apart from the notable exception of Ford (2007)'s structured streams (and the Click router, as it is natively designed for IP), all the solutions presented above require support in all the involved communication peers and, sometimes, networks. This makes them difficult to deploy in the already existing Internet infrastructure.

2.2.4 How to Best Implement Cross-Layering?

In small-scale controlled evaluations, cross-layer designs have shown encouragingly good results. However, only few of the proposals have been experimented within real systems, and some have not even been evaluated in realistic network simulations. Moreover, as is apparent for some of the work presented above, the context of enhanced operation of cross-layer designs tends to be very limited (*e.g.*, MANETs with specific routing and MAC protocols, or video streaming with selected application and physical layers). Overall, approaches crossing the layer boundaries tend to be very *ad hoc* solutions to very specific problems rather than readily usable in a wide range of contexts exhibiting similar problems.

The scarcity of general models of cross-layer interaction also limits the understanding of the implications of using such designs (Ci *et al.*, 2008) and leads to a concern that linking layers too tightly may hinder the applicability of the modified protocols. Kawadia and Kumar (2005) argue that, by modifying the behaviour of some layers, even if to improve them, one runs the risk of breaking some other layer's protocol assumptions. Taking as examples the proposals of ElBatt *et al.* (2000) and (resp.) Holland *et al.* (2001), they highlight their point by showing bad interactions with TCP's control loop and (resp.) a minimum hop routing algorithm. In both cases, the proposed enhancements conflict with the expectations of the other protocols and lead to worse performance than if using a purely standard approach.

Selected research works have used more analytical approaches to better model cross-layer designs and information sharing. Hortos (2003) recognises that metrics at one layer are really relevant to another only when *mapped* through the intermediate layers, *e.g.*, physical-layer modulation impacts the link-layer data rate, which in turn affects the transport-layer throughput and, ultimately, the application goodput in various ways depending on the protocols in use. Ci *et al.* (2008) make a similar observation in the limited context of multimedia applications. However, Grunenberger (2008, chap. 6, sec. 1.1) notes practical issues like mismatches in units, ranges or even meaning of some metrics reported by different hardware drivers. These discrepancies need be addressed before any mapping-based adaptation can be done. We more specifically review how decisions are made based on such metrics in the next section, and present some of the important metrics in Section 2.4.

Based on this review, it appears that implementing cross-layer approaches based on direct communication between layers results in too specific solutions which are unlikely to be of use in other contexts. Conversely, the new abstractions offer the largest freedom in information access, and the most control over the decisions. However, only a small number of these solutions are incrementally deployable. Therefore, it seems that the shared database approach, which introduces a vertical plane for cross-layer information exchange and decision may be the most practical approach. This may explain why this solution has already seen some standardisation work.

2.3 Decision Making

While cross-layer designs provide a wider possibility for information sharing and all the overviewed mechanisms enable some kind of action to be taken, we have purposefully omitted the discussion on how decisions to take such actions can be made based on the available information. It is the purpose of this section to summarise decision techniques, both those previously mentioned it this chapter and the relevant others which can assist in achieving the “Always Best Connected” (ABC) case of Gustafsson and Jonsson (2003).

The decision can be on a number of aspects: when to hand-off from an access link, which link to handover to, how to distribute traffic in multihomed scenarios, how to best adapt higher layers, or any combinations of these choices. We review these aspects in the remainder of this section.

2.3.1 Network Hand-off and Selection

Decisions regarding network hand-off and network selection are very similar tasks, as both need to evaluate the relevant access link (and network) based on some criteria to either decide to disconnect from, or connect to, this link. Both decisions are needed to complete a handover, but there is no requirement on their ordering: finding a more suitable network may be the trigger to start a hand-off but detecting that a currently connected network is no longer adequate (or even usable at all) can also activate the network selection mechanism. Both intra- (horizontal) and inter-technology (vertical) handovers are considered here. Yan *et al.* (2010) provide a more detailed survey and comparison of vertical hand-off decision mechanisms.

Criteria for Network Selection

A number of parameters have been introduced and used in the literature to discriminate access links and networks in order to select the best ones to connect to.

Radio signal quality The simplest mechanisms are based on measuring the quality of the radio signal (*e.g.*, SNR or RSS) and comparing it to a threshold for the hand-off decision, or selecting the network with the best value (Mäkelä *et al.*, 2000; Ylianttila *et al.*, 2001; Gwon *et al.*, 2002; Park *et al.*, 2003; Mohanty and Akyildiz, 2006).

Access link properties The threshold approach can also be applied to more precise metrics from the access link such as the delay or data rate (Wang *et al.*, 1999; Song and Jamalipour, 2005; Wilson *et al.*, 2005; Alkhawlani and Ayesh, 2008; Kandula *et al.*, 2008; Lahde and Wolf, 2009; Yahiya and Chaouchi, 2009).

Network usability Some networks only provide limited connectivity to the rest of the infrastructure or require specific credentials to grant access; the reachability of the current communication peers (Carter *et al.*, 2003) or the Internet in general (Nicholson *et al.*, 2006) are therefore obvious criteria to discriminate networks in such cases.

Network characteristics More relevant than the link layer properties for end-to-end communication facilitated by transport protocols like TCP, parameters such as network path capacities or RTTs are important to support feature-rich applications (Gazis *et al.*, 2005; Adamopoulou *et al.*, 2005; Nicholson *et al.*, 2006; Kandula *et al.*, 2008; Akyildiz *et al.*, 2005; Song and Jamalipour, 2005; Wilson *et al.*, 2005; Xing and Venkatasubramanian, 2005; Bari and Leung, 2007; Bonnin, 2008; Lee *et al.*, 2008; Zafeiris and Giakoumakis, 2008; Pang *et al.*, 2009; Yao *et al.*, 2009); some proposals also specifically take the application requirements into account in this phase (Liu *et al.*, 2006; Bari and Leung, 2007; Alkhawlani and Ayesh, 2008; Sun *et al.*, 2009).

Power consumption Not all networks or the underlying technologies can provide the same performance with the same power consumption; in a mobile context, battery life is important and trade-offs can be considered to preserve it (Xing and Venkatasubramanian, 2005; Rahmati and Zhong, 2007; Bonnin, 2008; Zafeiris and Giakoumakis, 2008; Sun *et al.*, 2009; Petander, 2009).

Access price Network operators have different pricing policies and though flat rates are not uncommon for fixed line access, mobile connectivity is still often priced on a data quantity quota base; it is therefore appropriate to take monetary considerations into account (Gazis *et al.*, 2005; Adamopoulou *et al.*, 2005; Bari and Leung, 2007; Alkhawlani and Ayesh, 2008; Bonnin, 2008; Zafeiris and Giakoumakis, 2008; Sun *et al.*, 2009; Piamrat *et al.*, 2010).

Infrastructure load An overloaded network is unlikely to provide good connectivity to prospective stations it is therefore the interest of both the user (Guo *et al.*, 2005; Liu *et al.*, 2006; Alkhawlani and Ayesh, 2008; Alperovich and Noble, 2010) and the network operator (Akyildiz *et al.*, 2005; Choque *et al.*, 2010) to establish new associations on the least contended links.

Application performance estimates The currently observed application layer performance, as observed by already connected nodes, can also be used as an indication of the "health" of a network link (Piamrat *et al.*, 2010).

Availability duration or mobility support To limit the number of handovers and their potential disruptions of performance, it may be relevant to select non-transient networks (Rathnayake and Ott, 2008), or those which allow some level of seamless intra-technology mobility compatible with the current mobility pattern of the node (Vidales *et al.*, 2005; Alkhawlani and Ayesh, 2008).

Non network-related context Finally, depending on high level descriptions of the current environment, some inferences can be made about which network would be the best (Yang and Galis, 2003; Rathnayake and Ott, 2008; Jeney *et al.*, 2009).

In addition, rather than measuring the relevant parameters, a number of works introduce some predictive estimates based on analytical models (Lee *et al.*, 2008; Choque *et al.*, 2010), other nodes' past observations (Pang *et al.*, 2009), history (Guo *et al.*, 2005; Rathnayake and Ott, 2008; Petander, 2009) or localisation (Rahmati and Zhong, 2007; Yao *et al.*, 2008; Jeney *et al.*, 2009; Yao *et al.*, 2009).

Multi-criteria Selection Techniques

To enable a finer selection of an access network, it may also be argued that considering a single criterion is not sufficient. Therefore, a number of more recent proposals use some sort of multi-objective optimisation (MOO) technique where the various criteria can be composed and compared.

Some of the most straightforward approaches select specific networks when some arbitrary conditions are met. These criteria are usually introduced based on external knowledge (Wang *et al.*, 1999; Chan *et al.*, 2001; Yang and Galis, 2003; Wilson *et al.*, 2005; Xing and Venkatasubramanian, 2005; Vidales *et al.*, 2005). Examples of such mechanisms include prioritising Wi-Fi over 3G access links—as most current smart-phones do—or giving a higher ranking to the least costly network. Other comparable approaches use utility functions in order to create a weighted compound variable for each network, to be compared to a threshold or that of other networks (Aust *et al.*, 2005; Liu *et al.*, 2006). Psaras and Mamatas (2011) balances guests of User-Provided Network (UPN; Sofia and Mendes, 2008) by probabilistically moving them to specific UPN-APs depending on their respective load. Both Song and Jamalipour (2005) and Chan *et al.* (2001) use the analytic hierarchy process (Saaty, 2000, AHP;), a more formal way of ranking choices, according to multiple criteria, to find the highest ranked options.

A number of proposals introduce sub-optimal but computationally efficient algorithms (Adamopoulou *et al.*, 2005; Xing and Venkatasubramanian, 2005). Linear programming techniques have been proposed to find optimal solutions for specific formulations of the problem (Zafeiris and Giakoumakis, 2008; Choque *et al.*, 2010), but these cannot always be used for real-time decisions. Other approaches, comparing the available choices to an optimal target, have been elaborated based on TOPSIS similarity distances (Bari and Leung, 2007; Sun *et al.*, 2009) or statistical likelihood that the considered option is the best (Yahiya and Chaouchi, 2009).

Finally, weights or scaling factors are an important parameter in MOOs, as the input variables need be mapped to comparable ranges. Alkhawlani and Ayesh (2008) use a genetic algorithm to derive these weights while Song and Jamalipour (2005) rely on grey relational analysis (Liu *et al.*, 2011). This preprocessing is also often done using fuzzy logic approaches (Chan *et al.*, 2001; Guo *et al.*, 2005; Wilson *et al.*, 2005; Alkhawlani and Ayesh, 2008).

2.3.2 Flow Distribution

The distribution of application flows over multiple uplinks active at the same time could be seen as a superset of the network selection schemes just presented. Indeed, quite a few of the proposals generalise some of the presented approaches to make this selection for each flow instead of only once per node.

However, the flow scheduling problem has to accommodate additional constraints. Perhaps the most important limitation is that only the networks to which the device is associated can be used, some of which are mutually exclusive, *e.g.*, two networks reachable by a device with only one interface for this technology. Solutions for traffic distribution in mobile networks proposed in the literature therefore can be classified in two categories depending on the level of interaction of the flow scheduling mechanism with the network selection algorithm.

The first class of solutions applies traffic classification and load balancing approaches of more conventional wired technologies in a second phase, after network uplinks have been selected and established. Zhao *et al.* (1998), Ylitalo *et al.* (2003) and Tsukada *et al.* (2008) use simple policies, based on flows' destinations or port, to decide which network is the most appropriate. The proposal of Nicholson *et al.* (2010) in essence randomly distributes flows to network as they are created, but provides a simple API for an application to choose a specific network, while Kandula *et al.* (2008) and Yao *et al.* (2009) distribute the new flows to the network with the least relative load in terms of capacity. Thompson *et al.* (2006) schedule the flows over the available networks in order to minimise the maximal transmission time; they use a flow-size predictor and consider the RTT of the networks more than their capacity when scheduling small flows.

Approaches in the second class take a more holistic approach by performing network selection and flow distribution at the same time. This gives the opportunity to derive more adequate solutions to the current usage pattern. A number of solutions rely on knowledge—be it assumed or explicit—of the applications' requirements to select the network which most closely matches them (Gazis *et al.*, 2005; Suciu *et al.*, 2005; Bonnin, 2008; Bonnin *et al.*, 2009). These approaches however come at the cost of a larger solution space to search. To address this issue, Singh *et al.* (2010) modelled the problem as a Markov chain to leverage decision process techniques on those models to explore the solution space more efficiently. Meanwhile, Zafeiris and Giakoumakis (2008) use binary integer programming techniques on their constrained model for the same purpose. Both solutions allow for redistribution of flows in addition to the on-line introduction of new flows that all proposals support.

2.3.3 Parameter Adaptation

Most cross-layer designs reviewed in Section 2.2 (and more specifically Section 2.2.1) are introduced to support the adaptation of one layer's parameter to another's. As is commonly the case with such approaches, the adaptation mechanisms tend to be very *ad hoc* and embed a lot of assumptions about the operation of the involved layers and the surrounding system.

However, some—mostly in the vertical calibration class—tend to use more elaborate techniques such as stochastic modelisation (Hortos, 2003) or analytical search of a constraint-based space (Johansson and Xiao, 2006; Ci *et al.*, 2008). All these approaches share the common characteristic that they lead to consider a very large state space, which the chosen models and solving techniques allow to explore with some degree of efficiency and/or[5] confidence in the optimality of the solution.

Several PhD theses have also contributed to the corpus of MOO applied to cognitive networks. Thomas (2007) introduced the basic framework and classification of cognitive networks as well as utility function-based MOOs in a game-theoretic context, Friend (2009) used both multi-agent and Markov decision processes approaches, and El-Nainay (2009) mainly focused on the use of genetic algorithms.

2.3.4 What Scope Should the Decision Cover?

Only some of the flow scheduling approaches reviewed above present themselves as superset of the network selection problem. In the vast majority of the decision mechanisms the focus is on a single task and the adaptation of the parameters of the other layers is approached in a completely orthogonal manner.

However, it is clear that one decision (*e.g.*, starting a hand-off) also directly impacts all other aspects of the system (*e.g.*, current flows and the performance of the application they belong to). Therefore, rather than micro-optimising each layer, it would seem appropriate to model the network stack as a whole, and let a single decision algorithm take care of finding, or trying to approach, the best operating point.

Intuitively, the search space of such a solution could easily become quite large. Attention should therefore be paid to strictly limiting the parameters to what is necessary, and modelling the problem in such a way that exploratory algorithms can effectively rule portions of the search space out without having to exhaustively explore it.

[5]"And/or" is "'and' and 'or'" or "'and' or 'or'."

2.4 Operational and Performance Metrics

Up to now, we only referred to metrics in generic, unqualified terms. This section provides an overview of various metrics defined in the literature which are relevant both for the operation of network protocols, and the evaluation of their performance. In the networking community, metrics tend to be grouped in "quality of —," which Stankiewicz *et al.* (2011) review in detail and show how different quality metrics relate to each other, and how they are interpreted by different standardisation bodies. For the sake of simplicity, we choose to use the term "metric" liberally in this dissertation, even for measures for which the triangle inequality may not be valid.

We first discuss network-level QoS, then present user-level quality of experience (QoE). We also mention some less usual criteria and metrics which can prove useful in our context, and conclude this section with an overview of the tools which can be used to measure these metrics.

2.4.1 Quality of Service

The concept of *quality of service* (QoS) encompasses all the metrics that can be used to describe the performance of a network segment or the full path between two end-points. Boundaries on QoS metrics are usually contractually agreed on between each customer and their network operator, who supports them by instantiating *classes of service* in which the customer's traffic is categorised. Technologies and protocols to enforce those boundaries are reviewed by Stankiewicz *et al.* (2011).

With respect to the definition of the metrics themselves, the IP Performance Metrics working group (IPPM)[6] at IETF has published several requests for comments. The existence of different concepts and definitions by other standardisation bodies, notably ITU-T recommendations, are also acknowledged in these documents. We choose to use the metrics defined in these RFCs for the remainder of this thesis. We argue that the clarity of the definitions and the coherence of the framework (Paxson *et al.*, 1998) justifies our choice. We summarise these metrics below.

Capacities and Throughput Chimento and Ishac (2008) note that several different capacities can be observed either at a link or path level, and at different layers. Their relation is defined as follows. We however took some liberty in slightly adapting the notations to something we believe is more discernible and usable.

link capacity $\mathrm{C}_L(l,t,i)$ is the average number of IP bits, including the header, that can be transferred, per time unit, over link l during period $[t, t+i[$. It depends on the physical layer's current *data rate* and the size of the preambles and MAC header.

link usage $\mathrm{Used}(l,t,i)$ is the actual number of IP bits transferred during period $[t, t+i[$.

link utilisation $\mathrm{Util}(l,t,i) = \mathrm{Used}(l,t,i)/\mathrm{C}_L(l,t,i)$ is the ratio of the link which is used during period $[t, t+i[$.

[6] http://datatracker.ietf.org/wg/ippm/

Table 2.3 First decimal and binary multiples of a byte, as defined by IEC 80000-13:2008.

Decimal multiples (SI)			Binary multiples		
Name	**Symbol**	**Value**	**Name**	**Symbol**	**Value**
Kilobyte	kB	10^3 B	Kibibyte	KiB	$2^{10}\,\mathrm{B} = 1.024\,\mathrm{kB}$
Megabyte	MB	10^6 B	Mebibyte	MiB	$2^{20}\,\mathrm{B} \simeq 1.049\,\mathrm{MB}$
Gigabyte	GB	10^9 B	Gibibyte	GiB	$2^{30}\,\mathrm{B} \simeq 1.074\,\mathrm{GB}$
			$\vdots$		

available link capacity $\mathrm{AvailCap}_L(l,t,i) = \mathrm{C}_L(l,t,i) \cdot (1 - \mathrm{Util}(l,t,i))$ is the additional average number of IP bits per unit of time that can be sent over link l during period $[t, t+i[$.

path capacity $\mathrm{C}_P(p,t,i) = \min_{i\in[0,n]} \mathrm{C}_L(l_i,t,i)$ is the available average link capacity along a path p composed of links $l_0, \dots, l_n$.

available path capacity $\mathrm{AvailCap}_P(p,t,i) = \min_{i\in[0,n]} \mathrm{AvailCap}_L(l_i,t,i)$ is the average number of IP bits, including the header, that can be transferred per time unit over path p during period $[t, t+i[$.

All capacities are measured in bits/s (or bps), while the link utilisation is a percentage.

In addition to these definitions, we use the terms *throughput* to refer to the capacity used by the transport protocol, and goodput for that made available to the application by the transport protocol. The throughput is only dependent on $\mathrm{AvailCap}_P(p,t,i)$ and the behaviour of the congestion control algorithm, while the goodput also depends on the size of the transport protocol's header. Taking a notation similar to Chimento and Ishac (2008)'s, the throughput $\mathrm{Thr}(p,t,i)$ therefore represents the average number of bits, including its own headers, that a transport protocol manages to send per unit of time during period $[t, t+i[$, and the goodput is that of data units (including application headers, if any) that the application manages to send in the same time. Throughput and goodput are also measured in bps, but do not include the IP headers. The latter also ignores the size of the headers of the transport protocol. We propose a formula for $\mathrm{Thr}(p,t,i)$ as (3.4) on page 54.

In this dissertation, we will use the above-defined term "capacity" in place of the commonly-used "bandwidth." Indeed, the latter has multiple overlapping meanings (either part of a spectrum, or related to data rate) in the field of wireless networks.

Finally, there tends to be some uncertainty with respect to multiples of base units when it comes to capacities in bits or bytes. It is usually unclear whether a kilobyte is 2^{10} or 10^3 bytes. IEC 80000-13:2008 standardises these notations by reasserting the SI notation (*e.g.*, k, G) to refer to powers of 10 and ratifies the binary multiples names and symbols (*e.g.*, Ki, Gi). This dissertation strictly adheres to these definitions, which are exemplified in Table 2.3.

One- and Two-Way Delays Packet delivery time—from the first bit of a packet being sent to the last one being received—are an important criterion for real-time and other delay-sensitive applications. However, as Almes *et al.* (1999a) note, it is difficult to obtain accurate one-way measurements unless the environment is tightly controlled, as a lot of factors can bias the observations (*e.g.*, lack of time synchronisation). They therefore introduce a formal definition of the RTT in (Almes *et al.*, 1999c). This two-way delay measurement is not as accurate—mainly as it may be the sum of the one-way delay of two different paths—but it is much more easily measured with precision, as only one node is in charge of all the measurement and calculation tasks. Both delays are measured in seconds.

Delay Variation and Jitter Demichelis and Chimento (2002) introduce a metric measuring the variation in the one-way delay. This variation is defined as the signed difference between the delay of two packets of the same length selected by some arbitrary function. As this metric is based on the variation of transit times, rather than the immediate values, it is more robust to loose time synchronisation between nodes than the one-way packet delay. The delay variation is also mesured in seconds.

Furthermore, the authors accept the *jitter* to be a specific instance of that metric where the selection function chooses consecutive packets and the absolute value of the difference is reported. Schulzrinne *et al.* (1996) apply an exponential filter with $\alpha = 1/16$ to this metric to obtain a time-varying estimate of the current jitter. It is computed at packet p as

$$\begin{aligned} \tau_p &= T_p^{Rcv} - T_p^{Snd} \\ \Delta\tau_p &= |\tau_p - \tau_{p-1}| \\ J_p &= J_{p-1} + \frac{1}{16}(\Delta\tau_p - J_{p-1}), \end{aligned} \tag{2.1}$$

where the T_p^* are the transmission and reception times of packet p.

Errors and Losses A lost packet is pragmatically defined by Almes *et al.* (1999b) as a packet which has not been received without error within a given—large—time period. This definition therefore covers losses due to congestion or contention as well as bit errors which may alter the packet and render it useless. The document does not explicitely mention the case of transport protocols which allow a certain level of corruption such as UDP-Lite (Larzon *et al.*, 2004) or DCCP's partial checksum coverage (Kohler *et al.*, 2006b); we assume "corrupted" to mean "refused by the destination." This metric is binary.

Almes *et al.* (1999b) then derive a Poisson-based sampling method to actively measure one-way packet losses along a network path without disrupting it. When the measurements are done in-band, *e.g.*, as part of the transport protocol's messages, it is however possible to sample all packets. Based on these samples, a more relevant metric, the packet loss average, can be derived. It is a percentage of lost packets, and its precision is directly linked to the sampling rate.

2.4.2 Quality of Experience

The QoS metrics presented above are purely network engineering criteria. They do not reflect the satisfaction of a user or the performance of an application. For this purpose, the concept of *quality of experience* (QoE) has recently raised a lot of interest (Kilkki, 2008). It is defined by ITU as "the overall acceptability of an application or service, as perceived subjectively by the end-user" (ITU-T Recommendation P.10/G.100 Amendment 2).

QoE is a complex measure which is dependent on many factors such as the achievable QoS, users' expectations or terminal and application characteristics, as represented in Figure 2.5. This section first reviews the scale and the evaluation processes proposed by ITU-T, then details the analytical models for common application and content types. It also covers some non content-related criteria which can impact the overall perceived quality.

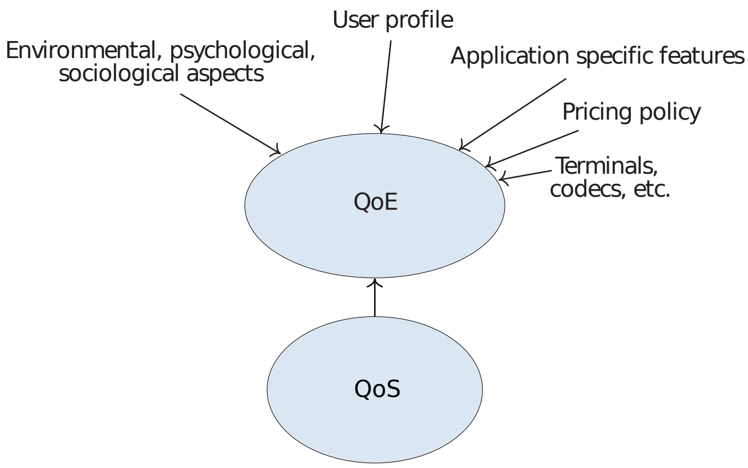


Figure 2.5 Quality of experience is a user-visible metric dependent on the performance of the underlying systems and various other contextual metrics (adapted with permission from Stankiewicz *et al.*, 2011).

Measuring Subjective QoE with the Mean Opinion Score

QoE is measured using the mean opinion score (MOS; ITU-T Recommendation P.800). MOS is an empirical quality scale which ranges from 5 (excellent) to 1 (bad). It is commonly accepted that MOSs under 3 (fair) are not tolerable. MOS was first introduced to allow for subjective evaluation of the quality of voice conversations in telecommunication systems but was later extended, as presented in the next section.

In the same document, ITU-T provides detailed recommendations on how to conduct large scale opinion tests to evaluate the MOS for a given set of parameters of a telecommunication system. These parameters include, most notably, the characteristics of the codec, the QoS of the transporting network, the properties of the terminal equipment, and the environment. The recommendation also includes directions for calibration of the test and the procedures to correctly collect MOS samples in these tests.

Objective Computational Models of QoE

Voice Conversation In several subsequent documents, ITU-T develops the E-model, a set of computational relations allowing to predict the expected QoE of various media based on information about the communication paths and equipments. ITU-T Recommendation G.107 analyses the mouth-to-ear path through the telecommunication system and introduces the R *factor* to assess the expected quality of a given connection,

$$R = 93.193 - I_s - I_d - I_{e-eff}, \tag{2.2}$$

where I_d is the one-way delay and echo impairment, I_{e-eff} accounts for the audio codec and its robustness to losses, and the packet loss rate. I_s is a combination of all impairments which occur simultaneously with the voice signal. Based on studies following the MOS assessment procedures (ITU-T Recommendation P.800), a cubic relationship between the R factor and MOS is defined as

$$MOS = 1 + 0.035 \cdot R + R(R-60)(100-R) \cdot 7 \times 10^{-6}. \tag{2.3}$$

Image and Video Winkler and Mohandas (2008) review techniques to measure the intrinsic quality of a video transmission at runtime. Perhaps the most common measure is the peak signal-to-noise ratio (PSNR; ANSI T1.TR.74-2001, 2001). PSNR is usually used to evaluate the quality of an $n \times m$ image I, or a sequence thereof, based on a reference R of the same dimensions. It allows to measure the amount of noise introduced by various treatments (*e.g.*, codec, lossy compression or transmission), by evaluating the mean square error,

$$\begin{aligned} MSE &= \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(R(i,j) - I(i,j)\right)^2, \\ PSNR &= 20 \log_{10} \frac{MAX_R}{\sqrt{MSE}}, \end{aligned} \tag{2.4}$$

where MAX_R is the maximum value that a pixel of R can take. PSNR is measured in decibels, with $+\infty$ being that of the reference itself (*i.e.*, $MSE = 0$).

Video-Telephony ITU-T's E-Model was also extended to video-telephony applications in ITU-T Recommendation G.1070. In order to integrate audio and video quality, the delay is decoupled from the rest of the R factor. This leads to the delay-independent metric

$$Q = 93.193 - I_{dte} - I_{e-eff},$$

in which only the echo I_{dte} is considered. The speech quality MOS S_q is derived from Q in the same way as (2.3). Similarly, a delay-less formula is proposed for the MOS of the video component,

$$V_q = 1 + I_{\text{coding}} \exp\left(\frac{P_{plV}}{D_{PplV}}\right), \tag{2.5}$$

where I_{coding} is the basic quality of the video codec for a given bit-rate and frame rate, while D_{PplV} is its loss robustness, and P_{plV} is the current loss rate. Finally, time is reintegrated in the model as a linear weighted combination of S_q, V_q and the respective delays.

IP-Based Elastic Traffic ITU-T Recommendation G.1030 provides an equivalent model for more general IP-based applications. It includes a regression-based estimate of the quality of web-browsing sessions,

$$\begin{aligned} MOS_{\text{web}} &= 5 + 4 \cdot \frac{\ln(WeightedST) - \ln(Min)}{\ln(Min) - \ln(Max)}, \\ WeightedST &= 0.54 \cdot T_1 + 0.72 \cdot T_2 + 0.98 \cdot T_3 + 1.76 \cdot T_4, \end{aligned} \tag{2.6}$$

where T_1 and T_2 are the times to display the first and, respectively, last element of a search page, and T_3 and T_4 are the corresponding times for the results page. Min and Max are the minimal expected and maximum accepted times for completion of the request.[7]

External Factors

The objective models introduced by ITU-T rely on a knowledge of both the terminal and the network QoS, while PSNR only allows to evaluate the distortion of image data along the communication path from source to destination. However, as Figure 2.5 has shown, other factors also impact the QoE but are not taken into account in the above models.

To the best of our knowledge, measures and models of these criteria are yet to be proposed and integrated into QoE metrics. Until then we try to establish a list of such criteria which could influence the perception of quality when using a mobile terminal.

Battery life As seen in Section 2.3, the power consumption of a battery-powered device is not negligible as too high a draw could greatly limit the usability of the device for any purpose. An empty battery would obviously render the QoE null, and should be pre-emptively avoided.

Price Most cellular networks provide access for a fee. However, it is clearly the interest of the user to minimise their expenses and, comparing two otherwise similar transmissions in terms of the other quality criteria, the one which incurs a lighter cost will always be preferred.

Security Some transmissions cannot afford the risk of eavesdropping. However, such a task is easy when the communication medium is wireless, as various such technologies and networks may provide a varying level of encryption. Depending on the type of transmission, the user may have a preference for using a properly secured access network, or not have it at all.

[7] The weights here are taken for a user expecting a 15 s session time; however, ITU-T Recommendation G.1030 contains more weight groups for other contexts.

Trust Related to the previous concern about security, trust can be seen as a more generic criterion, in which some networks or nodes in, *e.g.*, *ad hoc* networks may be trusted to a different degree to provide the expected service (*e.g.*, packet delivery).

2.4.3 How to Measure and Report These Metrics?

Having identified the relevant metrics, in this section, we provide an overview of measurement and recording methodologies related to those metrics. In a distributed network environment partly composed of mobile devices, measurement may not necessarily take place where the samples are needed and can be utilised. Reporting and collection systems are therefore needed in addition to measurement tools.

Measurement

Many tools exist for the observation of raw network performance. Most of them fit in two classes: *active probing* and *passive monitoring*. Probing tools actively generate traffic to test the characteristics and resilience of network paths. Avallone *et al.* (2004) proposed a distributed Internet traffic generation tool (D-ITG; Botta *et al.*, 2007) and compared it to the state of the art at the time to find their application most accurately tests the capacity of a path. More recently, however, Kolahi *et al.* (2011) conducted a similar experiment to find that more recent versions of Iperf[8] outperform all other tools, including D-ITG. Both references include a comprehensive list of active network probing tools, details of which are outside the scope of this thesis.

Passive traffic monitoring in networking environments is usually done via wrapper libraries hiding the operating system's underlying API such as the BSD or Linux capture filters (McCanne and Jacobson, 1993; Insolvibile, 2001). Perhaps the most common library for this purpose is the libpcap, derived from work on the `tcpdump` command line tool.[9] A more recent library designed for the same purpose is the libtrace (Alcock *et al.*, 2010), which offers a broader range of input and output formats, and a more detailed API to access information and fields contained in the captured data.

As mentioned in Sections 2.1.1 and 2.4.1, many transport protocols need to maintain indicators of the QoS of the path they are using in order to efficiently control the congestion. As an example, both TCP and TFRC directly measure the RTT R. The latter also directly computes its current rate while the current rate of TCP can easily be derived, for non data-starved senders, from its congestion window as $\texttt{cwnd}/R$. Based on such observations, Mathis *et al.* (2003) proposed, in the form of the Web100 Linux kernel patch and utilities,[10] an extension which allows to expose the internal parameters of established TCP sockets, thus enabling passive monitoring of the performance of the local stack. Tirumala *et al.* (2003) proposed a modification to Iperf based on this patch which, though it is not a passive observation tool, allows to greatly reduce the period during which the network is probed. The Network Diagnostic Tool (NDT; Zekauskas, 2005) is another Web100-based

[8] http://iperf.sourceforge.net/

[9] http://www.tcpdump.org

[10] http://www.web100.org/; the Web100 project has recently been continued with the Web10G project at http://www.web10g.org/.

client-server application to test a network path and observe the transport-layer parameters. The interface to TCP's extended statistics has also been standardised at IETF (Mathis *et al.*, 2007), but only seems to have been implemented under Linux and Windows.[11]

Contrary to QoS measurements, QoE is much more complex to evaluate, and would usually need *ad hoc* tools or, better, user feedback. Objective metrics can however be implemented in generic tools. For example, the `compare` command line tool[12] from ImageMagick (Still, 2005) features computations of the image PSNR, amongst other image quality metrics.

Reporting and Collection

When observing a complex system, it is necessary to ensure collection of the multiple samples using clearly defined units and appropriate timestamping to allow for valid comparisons and correlation when analysing the data. However, many of the tools highlighted in the previous section implement their own formatting and local-only reporting methods. In addition, various idiosyncrasies of the different tools[13] make it non-trivial to do even the simplest metric comparisons. Often, preprocessing is mandatory before any of the data can be used. This limits data re-usability in large-scale distributed measurements where observations from several tools on many network hosts must be correlated.

The Simple Network Management Protocol (SNMP; Harrington *et al.*, 2002; Harrington and Schoenwaelder, 2009) is an IETF standard providing a unique interface which can be used, amongst other things, to access operational and performance metrics of various network equipments. However, though the metrics which can be collected are clearly specified, they are limited to those included in the Media Information Base (MIB; Presuhn *et al.*, 2002). This base can be extended by local additions, but doing so limits the ability to share or reuse collected data, as the local definitions may not be as specific, or even available, as the standard MIB. Additionally, and more specifically for IP traffic, Claise *et al.* (2008) have proposed the IPFIX protocol to export flow information in a standardised format towards collection elements.

Several advanced frameworks have also been proposed to address the collection problem. PlanetFlow (Huang *et al.*, 2006) and CoMon (Park and Pai, 2006) provide flow logging and slice or node monitoring for PlanetLab (Fiuczynski and Matthews, 2006), including sophisticated query mechanisms; CoMo (Iannaccone, 2005) is a similar distributed flow measurement tool but not tied to PlanetLab. In a more mobile environment, Tsukada *et al.* (2010) proposed AnaVANET, a distributed traffic monitoring framework based on `tcpdump` and Iperf, which can perform *a posteriori* analysis of traces to correlate network performance to the localisation of each node in a VANET. Current evolution plans for the latter framework are considering the use of a real-time multi-sensor development platform (RTMaps SDK) to leverage this tool's accurately timestamped sample databases.

[11] http://kb.pert.geant.net/PERTKB/WebOneHundred

[12] http://www.imagemagick.org/script/compare.php

[13] As an example, Iperf can report transferred size and probed capacity either in bytes or bits, but without proper use of the IEC 80000-13:2008 prefixes, the meaning of the reported values is left as a guess for the experimenter. Inspection of the code reveals that powers of ten are used for bytes, while powers of two are used for bits.

MINER (Brandauer and Fichtel, 2009) and OMF Measurement Library (OML; White *et al.*, 2010) specifically focus on the instrumentation of applications to aggregate their measurements into central databases for convenient real-time or offline analysis. These tools are not tied to any specific type of measurement, framework or platform. They provide client libraries for use to extend any application with their unified measurement reporting channel. As such, these systems allow to leverage the large number of open-source applications by offering a simple way to report their measurements in a unified format to remote databases.

Though these proposals provide an experimenter with convenient reporting and collection tools, it is however unclear what accuracy or precision can be expected from either of them. Indeed, there are no studies that characterise these measurement collection tools and platforms in terms of their effects on the accuracy and precision of the measurements of the underlying systems they are helping researchers to observe. We address this question in Chapter 5 by evaluating the effect of OML instrumentation on selected applications.

2.5 Conclusion

This chapter has presented the state of the art of mobility mechanisms, cross-layer designs to better support disruptions and decision mechanisms to best take advantage of the multiplicity of wireless accesses. We have also presented performance metrics, both from a network device's and a user's perspective, as well as how they could be efficiently measured and shared in distributed mobile networks.

We have identified several issues which could benefit from further research. Namely, there are needs for decision mechanisms which can take a global view of the system into account, controllable network stack element to enforce the decisions, as well as reliable information exchange channels to support both internal adaptation based on currently observed performance, and external observation for performance evaluation. We develop these aspects in the following chapters.

Chapter 3

Multi-layer Optimisation of Network Choice and Usage

3.1 Introduction

Mobile devices, be they hand-held or on-board vehicles, increasingly support multiple interfaces, enabling them to connect to different wireless network technologies. Additionally, a number of service and network providers may also exist for any specific access technology, further increasing the number of available network choices. The concurrent availability of a number of networks presents both the opportunity and the problem of selecting the most appropriate ones. An ensuing problem is the distribution of application data flows over the selected networks for all the applications running on the mobile device. We call the problems of network selection and flow distribution the *multihomed flow management* problem. Also, we call *destination* of a flow the remote end-point of the connection, even if data is transferred from that correspondent peer to the mobile device.

Section 2.3 (page 32) reviewed the literature addressing the selection of appropriate access link and networks. It showed that the most common criteria include network quality of service (QoS) parameters such as capacity or delay. Researchers have also proposed more user-centric criteria which are highly relevant to a mobile user, such as the power consumption (Wang *et al.*, 1999; Petander, 2009), or the cost of network use (Wilson *et al.*, 2005; Alkhawlani and Ayesh, 2008). In some of the aforementioned, the user is also expected to provide policies or preferences (Wang *et al.*, 1999; Song and Jamalipour, 2005; Alkhawlani and Ayesh, 2008).

However, as Kilkki (2008) highlights, there is growing awareness of quality of experience (QoE) in the research community. This emerging concept casts a new light on the meaning of Gustafsson and Jonsson (2003)'s Always Best Connected (ABC) nodes. It seems that the question of whether a device is technically best connected should be replaced by the evaluation of whether the *end-user* feels their terminal provides them with the best currently achievable performance. Though we focus on this previous case, this question can also cover applications not directly visible to humans. We therefore propose that the multihomed flow management problem would be best solved by considering the criteria of application quality, mobile resource use and price of network service. We collectively refer to these as *high-level performance metrics*.

We consider application quality metrics directly, rather than by relying on the network QoS. We argue that the non-linear relationship between the application quality and QoS is a good motivation for this approach. Piamrat *et al.* (2010) already considered QoE in order to select the optimum access link. However, they use it only as a single global metric for that link, while we recognise quality can vary differently depending on the application, and should therefore be treated with a finer level of granularity.

Additionally, we concluded in Sections 2.2.4 (page 31) and 2.3.4 (page 36) that micro-optimisation is a risk in cross-layer designs, and that one had better designing decision mechanisms which span the entire stack. We therefore include adaptive variation of the application and protocol parameters in our proposal. By pre-emptively determining the best set of parameters, we can greatly shorten the adaptation process of the application and underlying transport protocol parameters, among others, to the network conditions.

In this chapter, we introduce a generic decision mechanism with global view and control of the network stack (Figure 3.1) in order to avoid the potentially adverse interactions risk presented by Kawadia and Kumar (2005). In Section 3.2, we start by describing the vision a multihomed device has of its environment. In Section 3.3, we formalise this description as a constrained optimisation model while Section 3.4 develops how we intend to use the quality metrics introduced in Section 2.4.2 (page 40). Section 3.5 presents scenarios in which we evaluated this proposal, with the evaluation itself and comparison with commonly used approaches presented in Section 3.6. We conclude and present future work in Section 3.7.

3.2 Environment of a Multihomed Mobile Device

We consider MIPv6-enabled devices with support for Multiple Care-of Address (MCoA). Each of those mobile nodes (MNs) can therefore inform their home agent (HA) and correspondent nodes (CNs) about their current locators (care-of address, CoA) to which the data flows are to be directed. A multihomed device has more than one network interface, and therefore multiple CoAs. No limitation is made on the technologies of these interfaces: they can be all similar, all different, or any combination. The MN can use each of these interfaces to connect to any compatible access link in range. We also consider that the MN can instruct how each data flow should be directed to their destination (Mitsuya *et al.*, 2007; Larsson *et al.*, 2009; Tsirtsis *et al.*, 2011). This multihomed device can use several access links and networks to support its communications with other servers and peers in the Internet, as shown in Figures 3.2 and 3.3.

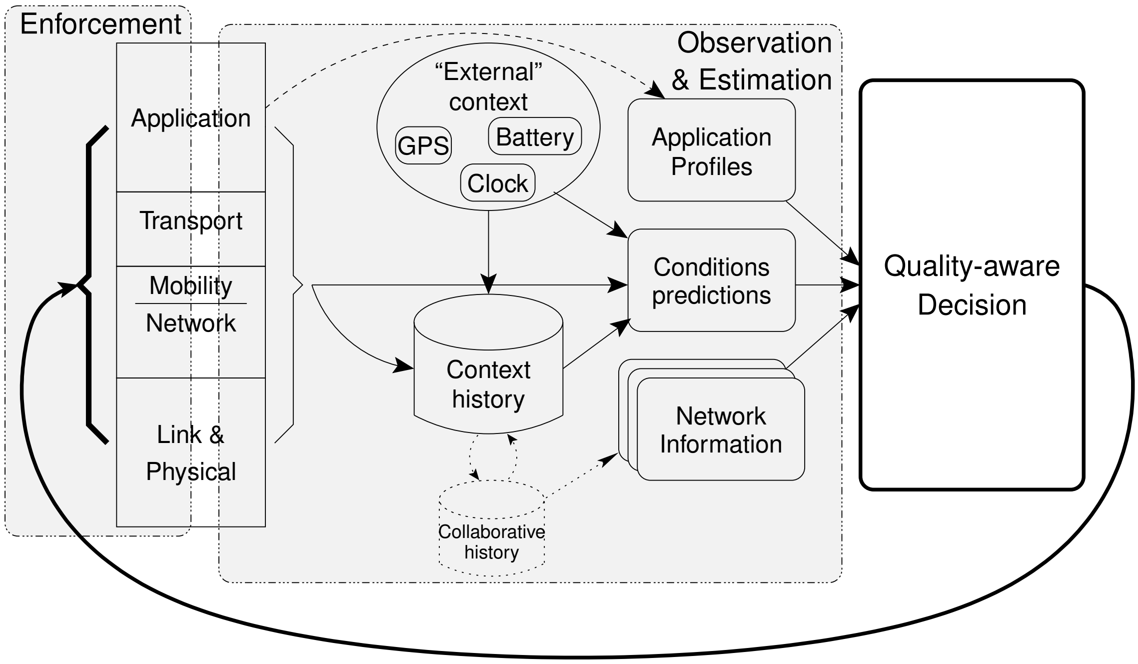


Figure 3.1 Relation of the contribution presented in this chapter to the cross-layer framework of this thesis. Bold lines and non greyed-out components are the current focus.

3.2.1 Partial View of the Network Path

Due to its location and interactions with the elements along each IP network path, an MN only has a partial view of its environment. It has full control and information over the local link to its first hop, usually via the link layer of the relevant network interface. However, forwarding decisions along the rest of network route are usually made locally at each router, and are beyond the control, or even knowledge, of the MN. Only aggregate information about the route can be observed and collected by this node. This information is usually collected and used at the transport layer. Figure 3.4 shows a decomposition of some paths of Figure 3.2 with respect to the knowledge and control of the MN.

We assume that a network n can have one or more access routers, or gateways, g, each providing one access network to the MN. The MN can decide to establish a link l to some of the gateways of the access networks. That link then represents the first hop of a data flow's packets forwarded for the MN. The network route r is an aggregate view of the rest of the way, from the gateway's operator's network to the flow's destination d. Regardless of which technology the MN uses to get its packets into the network, it is reasonable to assume that the rest of the route from any access router of a given network will have the same properties towards a given destination.

3.2.2 Conditions Along the End-to-end Path

An end-to-end path, from a local interface to a given destination, can be seen as a combination of the physical link from that interface to an access network and the route from that

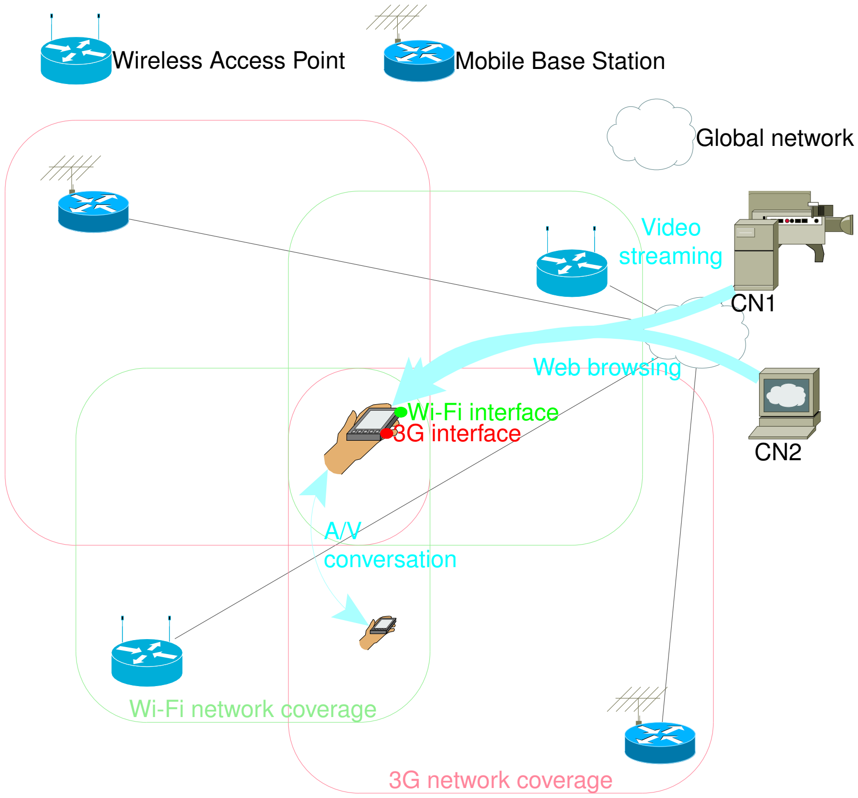
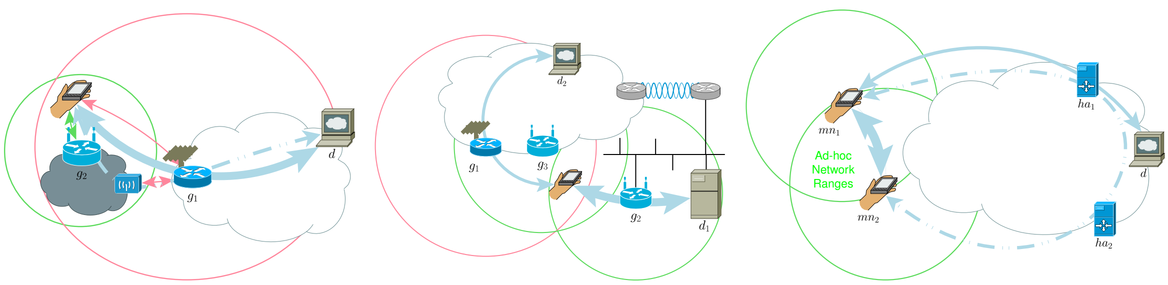


Figure 3.2 A multihomed mobile device can access multiple wireless networks of various types at the same time to carry flows to and from remote destinations.



(a) Shared uplink (b) Most direct connection (c) Shorter *ad hoc* path

Figure 3.3 Typical use-case scenarios of a multihomed mobile device: (a) Public Wi-Fi provisioned with a 3G uplink (*e.g.*, commuter train or ferry); (b) Data server on a network offering wireless access but a slow uplink; (c) 802.11 *ad hoc* link is available which is much shorter than going through both HAs or even the infrastructure.

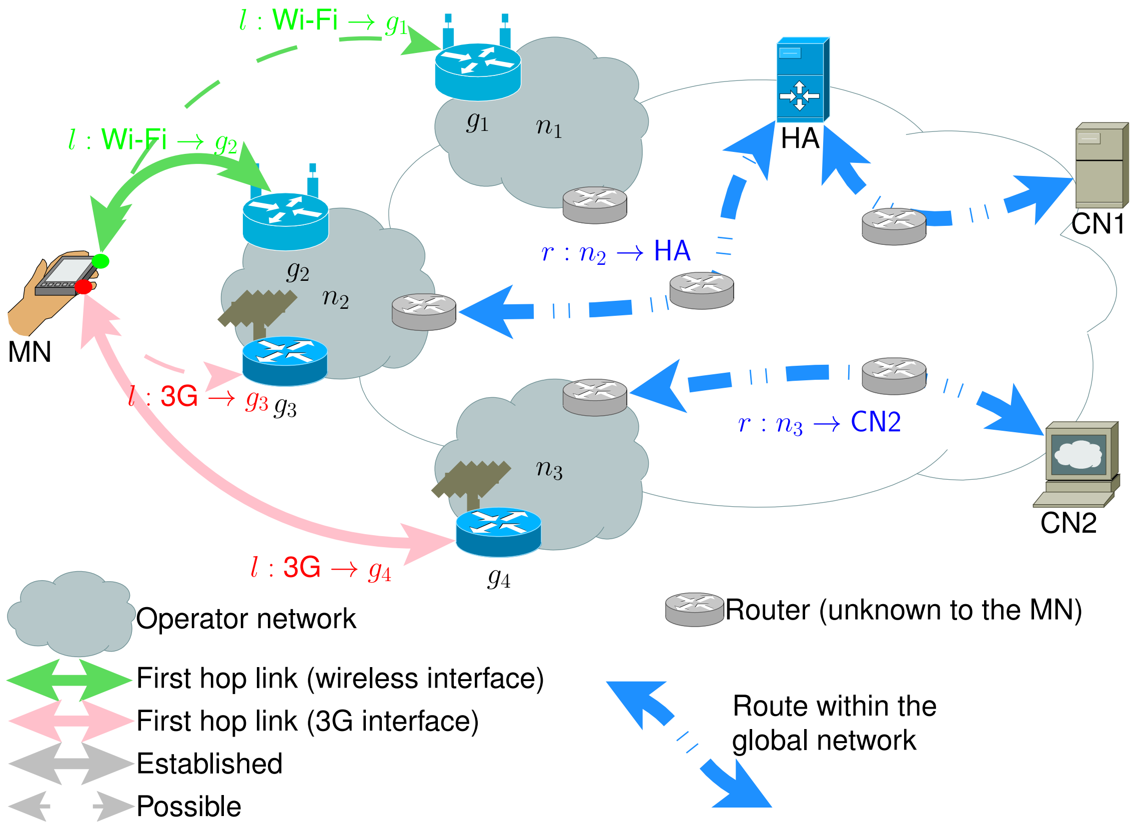


Figure 3.4 An MN has a partial view of the network path to its CNs. It has good information about the locally available access routers g_i accessible from its interfaces on the wireless media. Its view and control of the network route from the operators' networks n_j to which those routers belong is less precise and lacks detailed information such as other intermediate routers. Here, CN1 does not natively support mobility features and flows exchanged with this peer have to go through the HA. Conversely, CN2 supports these features and related flows directly use the MN's selected CoA.

access network to the destination. An additional element has also to be taken into account to determine the network performance an application can achieve along that path: the socket interface between the application and the network stack. The transport protocol may indeed induce some rate limitation and potential losses may happen with real-time application data. Figure 3.5 illustrates this decomposition.

First Hop Link

We consider direct links from the MN to a next hop using one of its interfaces. For the sake of simplicity of the discussion, we assume that a single network-layer datagram is encapsulated in each frame, and no fragmentation occurs (*i.e.*, each frame corresponds to one and only one complete IP packet). This section also lists the power consumption and monetary costs that may be incurred when using a specific link as additional metrics.

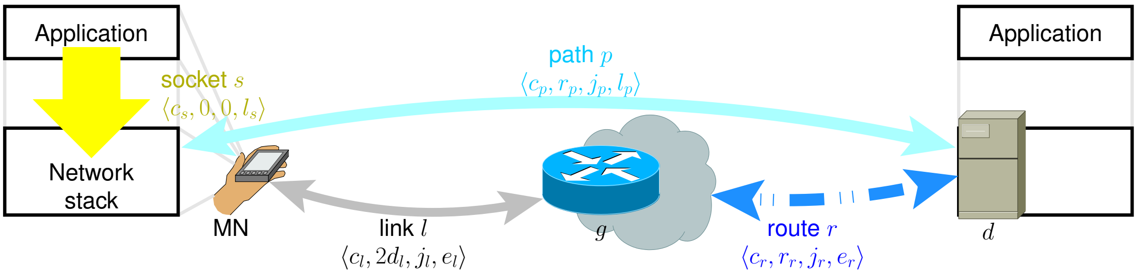


Figure 3.5 Three main elements contribute to the performance of network path p from the perspective of an application running on an MN: the first hop on the local link l, the route r from the operator network to the destination, and the socket interface between the application and the network stack s. Capacity, delays (RTT and jitter) and loss metrics are represented as a tuple for each element.

Capacity The physical and link metrics are highly dependent on the type of network interface and technology in use. All network technologies have a physical limitation on the maximal speed at which they can put symbols on the medium. This maximal capacity achievable through that interface is simply $\mathrm{C}_L(l, t, \cdot)$[1] as defined by Chimento and Ishac (2008). However, depending on the spectrum use and load of the gateways, the MN can only expect to get a capacity of $c_l = \mathrm{AvailCap}_L(l, t, \cdot)$ from that link to carry its flow.

Delays At the link layer, the medium access control (MAC) protocol allows to make the spectrum a reliable shared link. An interface taking part in the distributed MAC algorithm may delay sending frames until the medium is next free. Until then, packets to be sent are queued in local buffers. This introduces a delay in the delivery of those packets, and may even result in losses if the station has to contend for the network for too long while its buffers fill, or the packets' deadlines expire. The delay is denoted d_l, and the link's contribution to the RTT can be estimated as $2d_l$. In addition, as the most common MAC algorithms rely on random back-offs (*e.g.*, IEEE Std 802.11-2007), they also introduce some jitter j_l.

Packet Error and Loss Rates As outlined is Section 2.1.2 (page 17), wireless links are prone to experience packet losses unrelated to congestion. Packets sent properly, as far as the sending station is concerned, may also be decoded incorrectly due to transient interferences at the receiver. Such errors lead to the packet being discarded and counted as lost. This packet error rate is denoted e_l. MAC-level retransmissions can mitigate these losses, at the price of an increased delay and jitter.

Power Consumption Transmitting data over a wireless network, or merely keeping an interface on, draws power from the node's battery. When an interface i is uploading data at speed s_u and downloading at s_d, its instantaneous power consumption can intuitively

[1] We use notation "$\cdot$" in functional relations as placeholders for parameters which are not relevant to the current discussion.

be estimated as the sum of both components,

$$Pw(i) = Pw_a(i) + (Pw_u(i) \cdot s_u + Pw_d(i) \cdot s_d)\,, \tag{3.1}$$

where $Pw_a(i)$ is the power consumption of the interface when it is active and associated to a network, but and $Pw_u(i)$ (resp. $Pw_d(i)$) is that of using it to upload (resp. download) 1 bps.

Monetary Cost Some networks may not be free, and charge the user various types of usage fees. Those can either be for time access, transfer volume, or both. Thus, transmitting data at speed $s = s_d + s_u$ would cost

$$Pr(n) = Pr_a(n) + Pr_t(n) \cdot s, \tag{3.2}$$

where $Pr_a(n)$ is the cost of having an active connection to the access network n for a second, and $Pr_t(n)$ is that of using it to transmit 1 bps.

Network Route

The physical link directly connected to the mobile node is rarely the only one traversed by data flows before they reach their destination. They will be forwarded by a series of other links with varying characteristics. We refer to this part of a path as *network route*, as this is how it appears to the MN.

A network route r is intrinsically a network path in the acceptation of IP Performance Metrics working group (IPPM), presented in Section 2.4.1 (page 37), as it is composed of one or more links. From the perspective of the MN, r is all of the path to the destination d but the first hop-link l. A flow may however need to go through the HA if d does not natively support route optimisation (RO), which is likely to increase the length of the route (triangular routing).

Capacity Similarly to our approach for the first hop link, the achievable capacity of route r is the path capacity $c_r = \mathrm{AvailCap}_P(r, t, \cdot)$.

Delays As Almes *et al.* (1999a) noted, it is easier to observe the RTT r_r of a route from an access network to a destination. Under the assumption that a network path is symmetric for delays, the one way delay can be roughly derived by halving the RTT. The routers' queues along the route may however introduce a variable delay which will manifest itself as jitter at the destination, j_r.

Packet Loss Rates Properly maintained networks should not experience any loss. Losses native to the path infrastructure (*i.e.*, not related to congestions) are denoted e_r, but should usually be null.

Socket Interface

A socket is the programming abstraction for the connection as seen from one of its applicative endpoints. As such, it is the interface through which the application layer uses a network path, and is dependent on its performance. Being within the same machine, this link is highly unlikely to add significant delay to the transmission, but may limit the usage of the available capacity along a path (*e.g.*, congestion control); doing so, it may also create losses for the application.

Congestion-Limited Throughput As we consider shared networks, only congestion controlled transport protocols are discussed. As mentioned in Section 2.1.2 (page 17), wireless losses may disrupt such rate controls' performance. For TCP (and TFRC), it is possible to determine the maximal throughput that the application would be able to achieve under certain loss conditions. The model of TCP derived by Padhye *et al.* (1998) allows to derive an average stationary rate given the packet loss rate $l_p = e_l \cdot e_r$ and the RTT $r_p = 2d_l + r_r$ along the full path as

$$\mathrm{TCP}(r_p, l_p) = \frac{MTU}{r_p\left(\sqrt{\frac{2l_p}{3}} + 12\sqrt{\frac{3l_p}{8}}l_p(1+32{l_p}^2)\right)}, \tag{3.3}$$

where MTU is the maximum size of a packet (Maximum Transmission Unit). It is noteworthy to observe that baseline packet error rates of wireless networks, after MAC retransmissions, are sufficiently low so as not to limit TCP's throughput more than the nominal capacity of the access technology, as illustrated in Figure 3.6. This is, however, not always the case for higher loss rates. Equation (3.3) can therefore be used in a predictor of the expected throughput for a TCP or TFRC sender on path p,

$$\mathrm{Thr}(p,t,i) = \min\left(\mathrm{TCP}(r_p, l_p), \mathrm{AvailCap}_p(p,t,i)\right). \tag{3.4}$$

This measure is therefore the end-to-end capacity available to the application using socket s over path p,

$$c_s = \mathrm{Thr}(p,t,i). \tag{3.5}$$

However, this rate may be lower than the rate at which the application has to send its packets. In this case, the application will experience packet losses as the socket's buffer fills up but the transport protocol cannot send its contents fast enough.

Application Data Losses Though not all applications have a capacity requirement c_{req}, we propose to evaluate the loss rate l_s of those which do as

$$l_s = \begin{cases} (c_{\mathrm{req}} - c_s)/c_{\mathrm{req}} & c_{\mathrm{req}} > c_s, \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

when their throughput is limited by the congestion-control algorithm of the transport protocol in use. This is more specifically the case of real-time traffic when the load from the application cannot adapt without changes in the encoding parameters.

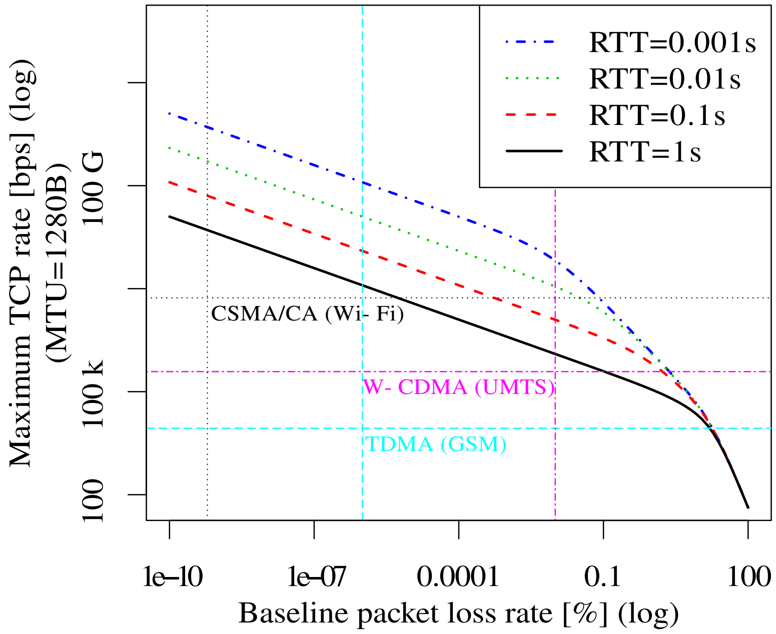


Figure 3.6 TCP's maximum throughput under various delay and loss conditions. Typical capacities and loss rates (after retransmissions) for TDMA (GSM) and CDMA/CA (Xylomenos *et al.*, 2001), and W-CDMA (UMTS; Prokkola *et al.*, 2009) are reported. They suggest that these errors alone may be ignored as a throughput-limiting factor for TCP.

3.2.3 Experimental Evaluation of Path Conditions

To obtain realistic figures for the characteristic conditions of a path from an application or user perspective, we set up a QoS measurement testbed, depicted in Figure 3.7. We also analysed data from external sources in order to evaluate the power consumption and price of using specific access networks.

End-to-end Path QoS

Our QoS measurement system is based on common network administration and research tools. The client software runs on mobile devices equipped with multiple interfaces for various network technologies. It collects information about the current path to a test server (*e.g.*, RTT, number and location of intermediate servers or presence of NAT), performs two-way throughput tests using Iperf and the command-line client Wget[2], and

[2] http://www.gnu.org/software/wget/

investigates other transport-layer parameters with the Network Diagnostic Tool (NDT) suite (Zekauskas, 2005).

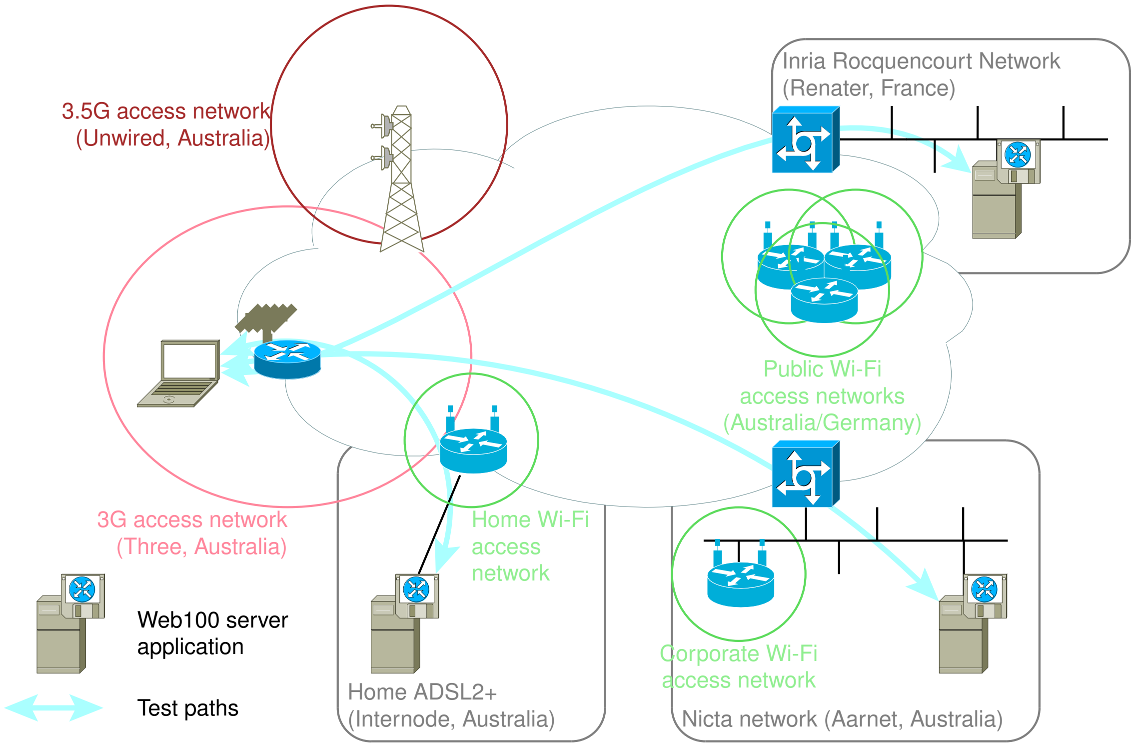


Figure 3.7 QoS measurement testbed. Several test servers are located in Australia and in France. The test clients connects from the locally available access networks (private or public Wi-Fi, 3G or pre-WiMAX) to the servers in sequence, and test the network path.

Due to the lack of control on the intermediate routers, already mentioned in Section 3.2.1, our system does not allow us to obtain detailed characteristics for all elements of Figure 3.5. Only c_s, the end-to-end capacity, can be collected through the use of either Iperf and Wget, while r_p is obtained through the use of the system `ping` tool. NDT can collect both metrics.

For this testbed, three networks hosted measurement servers: two academic networks in Australia and in France (Aarnet and Renater), and an ADSL2+ provided by an Australian ISP (Internode). Test clients were laptops running a Linux 2.6 kernel, equipped with a 3G modem with a subscription to the Australian operator Three and a pre-WiMAX technology deployment (3.5G) based on Navini Ripwave modems (operated by Unwired), in addition to their built-in Wi-Fi interface. Public hotspots as well as some private ones available to the experimenter have been used as access networks to conduct the Wi-Fi tests.

Over a period of a few months (September–November 2010), measurements using this testbed have been taken from several locations in Sydney, Australia, and Bremen, Germany. Only Wi-Fi tests were run at the latter location. Figure 3.8 summarises the measured characteristics for each network type.

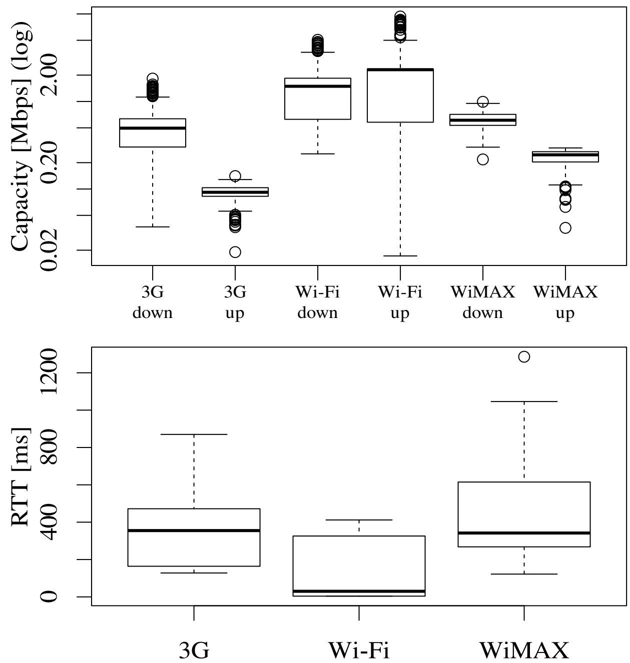


Figure 3.8 Empirical QoS distributions as measured from the testbed. Top: upstream and downstream capacity, bottom: RTT. Boxes show the 25–75% inter-quartile range (IQR), the heavier line represents the median of the data-set and whiskers extends to the extreme data within $1.5IQR$ from the median. Circles denote outliers.

Power Usage and Battery Consumption

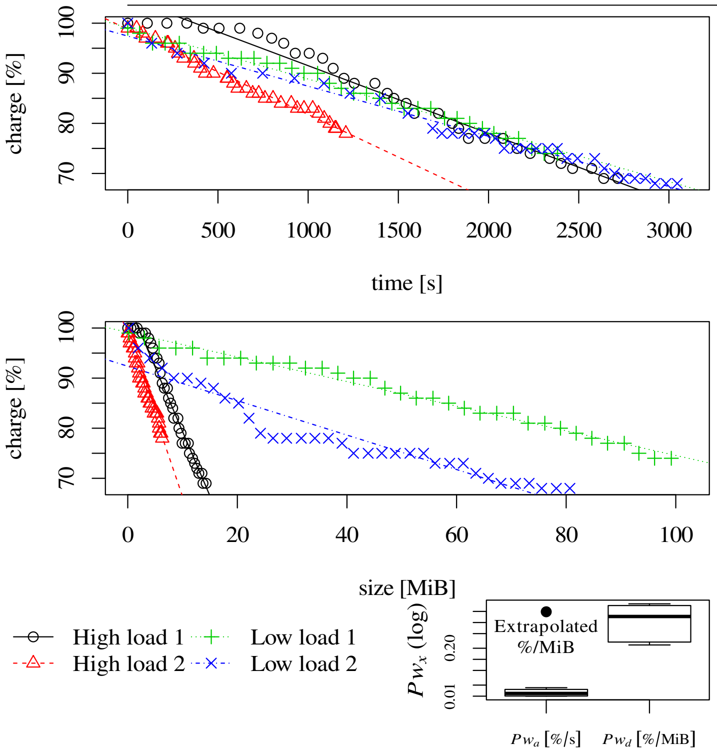
We obtained the 3G and Wi-Fi power consumption data-set used by Petander (2009). Figure 3.9 shows characteristic examples from the dataset: UMTS uploads and, Wi-Fi downloads and idleness. We analysed it using linear models to evaluate the impact of time and transmission speed on the battery. Table 3.1 summarises the parameters and goodness of fit (in the form of residual standard errors, RSE) of regressions of dependent variable battery consumption against independent variables active time (for $\overline{Pw_a}$) and amount of transferred data ($\overline{Pw_{d,u}}$).

This analysis confirms that the main drain comes from whether the radio circuit is powered. The transmission speed does not seem to have a strong impact on the power consumption, and is more dependent on the signal quality and network load. $Pw_a(i)$, in (3.1), is a much more important factor than $Pw_d, u(i)$. Equation (3.1) can therefore be simplified to

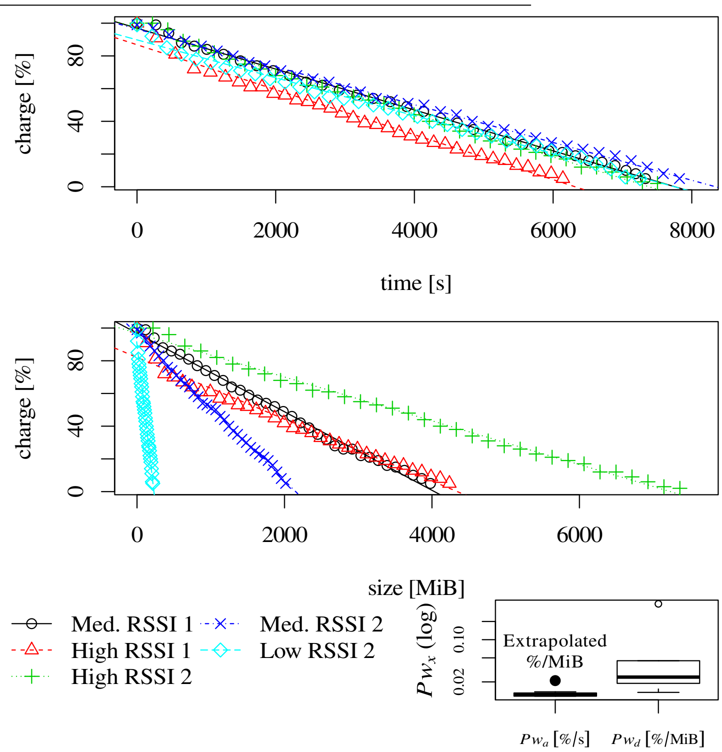
$$Pw(i) \simeq Pw_a(i). \tag{3.7}$$

Monetary Cost of Using an Access Network

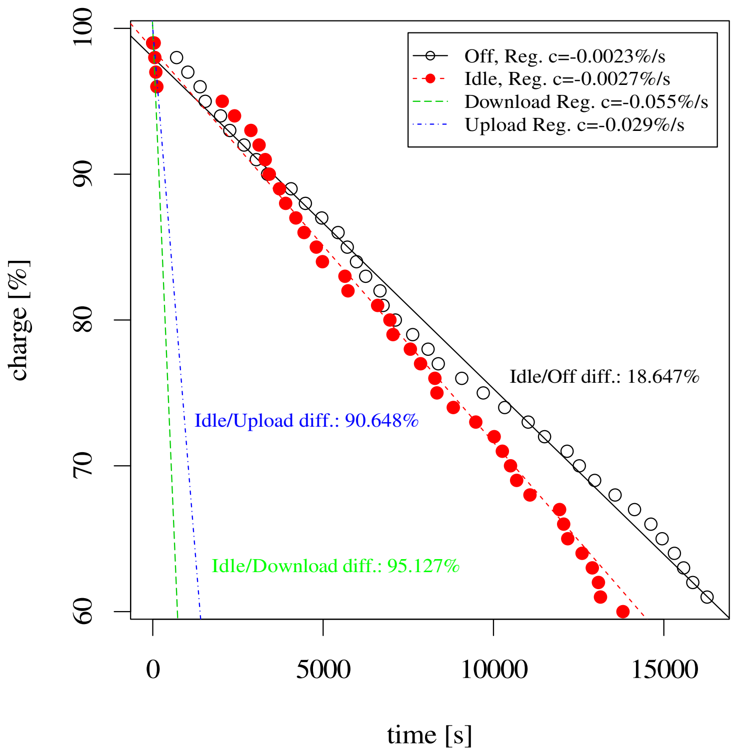
The pricing terms of the main mobile broadband operators in Australia were collected in December 2010. There is a wide variety of contract types (timed or quota, plan or prepaid,



(a) UMTS upload battery usage (high load).



(b) Wi-Fi download battery usage.



(c) Wi-Fi battery usage when idle.

Figure 3.9 Empirical battery consumption of UMTS and Wi-Fi devices in Petander (2009)'s dataset, and linear regression against time and transferred size. Active interfaces exhibit a time-constant power consumption, regardless of the amount of data transferred. Inactive but enabled Wi-Fi interfaces also incur a significant battery usage. For legibility, not all empirical data points are reported.

Table 3.1 Averages and residual standard errors (RSE) of the regressions performed on the battery dataset. Fitting with only Pw_a yields much smaller and less varying RSEs than for $Pw_{d,u}$.

Technology	Direction	$\overline{Pw_a}$	$SE_{\overline{Pw_a}}$	$\overline{RSE_a}$	$SE_{\overline{RSE_a}}$	$\overline{Pw_{d,u}}$	$SE_{\overline{Pw_{d,u}}}$	$\overline{RSE_{d,u}}$	$SE_{\overline{RSE_{d,u}}}$
UMTS high load	download	0.017	0.0018	1.35	0.20	1.31	0.34	2.26	0.36
	upload	0.013	0.0013	1.04	0.16	1.59	0.59	1.48	0.33
UMTS low load	download	0.013	0.0022	1.87	0.34	0.24	0.0079	3.12	1.37
	upload	0.011	0.0010	1.61	0.014	0.27	0.023	1.60	0.12
Wi-Fi	download	0.012	0.00034	1.96	0.22	0.098	0.060	2.32	0.34
	upload	0.015	0.00050	2.39	0.38	0.042	0.0067	2.81	0.42

peak and off-peak periods) and ways to handle excess usage (increased price, data blocks, traffic shaping).

Depending on commercial factors such as bundled plans, quotas or off-peak period, Pr_a and Pr_u see large variations depending on the users' operator and contract o_c, history of usage h and time of the day t. A more realistic formulation of (3.2) for the monetary cost of using a network should therefore be

$$Pr = Pr_a(o_c, h, t) + Pr_t(o_c, h, t)s. \tag{3.8}$$

However, at any point in time T, $Pr_* = Pr_*(o_c, h, T)$ can be precomputed, leaving (3.2) as a good approximation of the price of using an access network.

3.3 Multihomed Flow Management

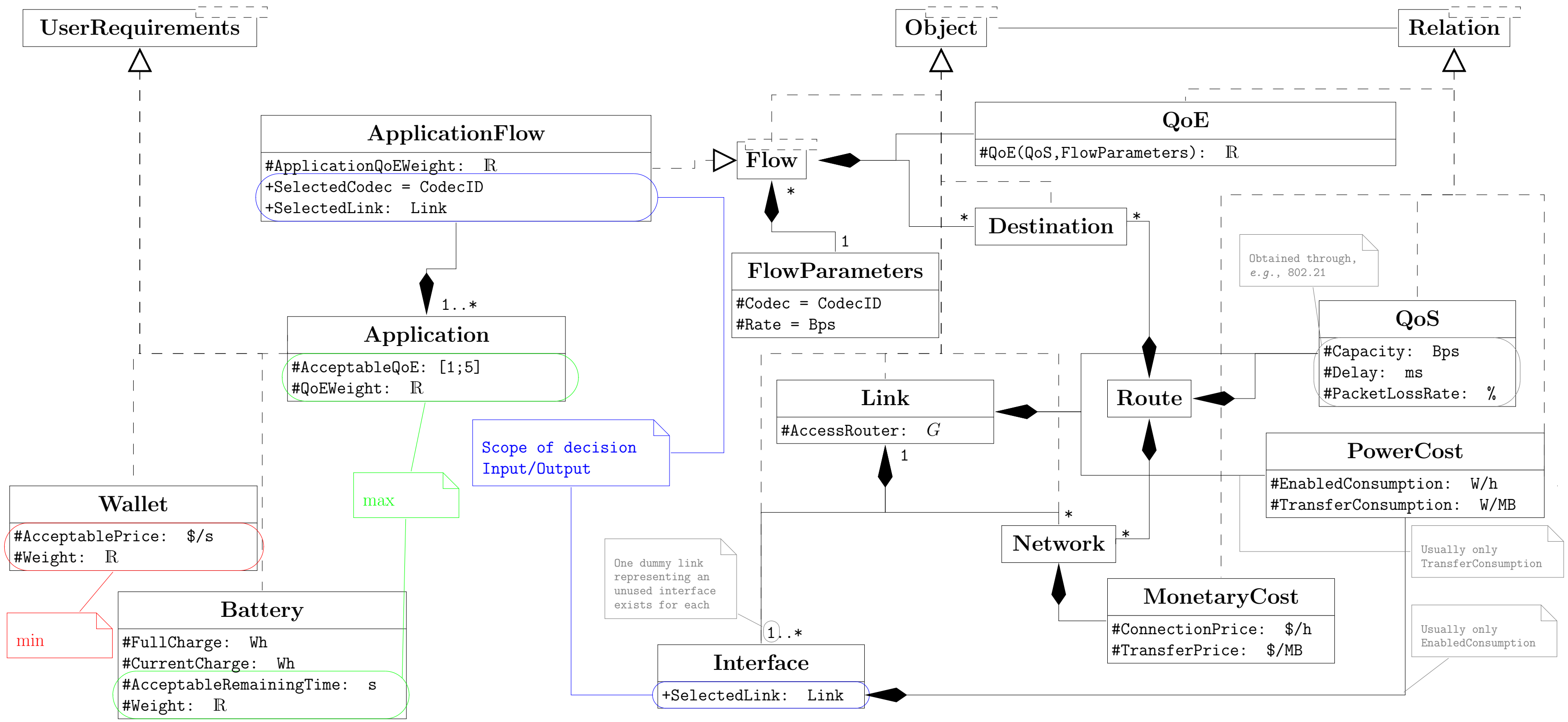
We propose to optimise overall performance metrics by deciding on network associations, distribution of flows across links and application parameters. The UML model shown in Figure 3.10 is proposed as a summary of the basic interactions between conditions presented in the previous section and some optimisation objectives.

We model this proposal as a constrained optimisation problem (COP). This section presents the model, the objective function to be optimised, and the structural constraints. The sets and operations are summarised in Table 3.2.

3.3.1 The Flow Management Problem

Let I be the set of network interfaces, N the set of all available networks, including the special network None, and $L \subseteq I \times N$ the set of links that can be established between interfaces and networks. At all times, each interface $i \in I$ is associated to a network $n \in N$. This is represented as link $l_i = (i, n)$. Vector $\mathbf{a}$ of length $|I|$ represents the network associations of all the interfaces $i \in I$; $a_i = n$ when $l_i = (i, n)$.

60



Constraints:

Associations' multiplicity is 1 unless otherwise stated.

$$\sum_{f \in F | f.\mathtt{SelectedLink}=l} \mathrm{FlowParameters}(f.\mathtt{SelectedCodec}).\mathtt{Rate} \leq l.\mathtt{Capacity}\, \forall l \in L$$

$$\sum_{f \in F | Network(f.\mathtt{SelectedLink})=n, Destination(f)=d} \mathrm{FlowParameters}(f.\mathtt{SelectedCodec}).\mathtt{Rate} \leq n.\mathtt{Capacity}\, \forall n \in N,\, d \in D$$

Figure 3.10 Environmental elements involved in the multihomed flow management problem.

Table 3.2 Sets and their operations used to define and model the multihomed flow management problem.

	Set of access networks N
None $\in N$	null network to represent unassociated interfaces
	Set of interfaces I
$\mathbf{a} \in N^{\lvert I \rvert}$	network association vector where $a_i \in N$, $\forall i \in I$
	Set of links $L \subseteq I \times N$
$QoS(l)$	achievable QoS achievable on link $l \in L$
$Pw(l)$	power consumption of link l, Eq. (3.7)
$Pr(l)$	access price of link l, Eq. (3.2)
	QoS tuple $q = \langle c, r, e, s, \ldots \rangle$
$C(q) = c$	available capacity
$R(q) = r$	round-trip time
e	link error rate
s	security level
$\ldots$	other metrics relevant to an application
	Set of flows F
$\mathbf{d} \in L^{\lvert F \rvert}$	flow distribution vector where $d_f \in L$, $\forall f \in F$
$\mathbf{p}$, $\lvert \mathbf{p} \rvert = \lvert F \rvert$	application-specific parameters (p_f for flow f)
$Q(f, p_f, q_f)$	quality profile of flow $f \in F$ under QoS q_f
$q_{\text{req}}(f, p_f)$	min. required QoS to maximise $Q(f, p_f, q_{\text{req}}(f, p_f))$

We define operation $QoS(\cdot) = \langle c, r, e, s, \ldots \rangle$ on elements of L which represents the QoS achievable on a given link. Components of $QoS(l)$ include the capacity of the link $C(l) = C(QoS(l)) = c$, the round-trip time $R(l) = R(QoS(l)) = r$ and the potential error rate e. It can also include indices such as the security level s (*e.g.*, WPA2 would rank better than WEP). We also define two other operations on links, the induced power consumption $Pw(\cdot)$ and the price that some operators charge for use of their network, $Pr(\cdot)$. These operations are the respective equivalent of (3.7) and (3.2).

Additionally, let F be the set of application flows which have to be distributed on active interfaces. The performance quality of a flow f can be expressed as $Q(f, p_f, q_f)$ where p_f is a set of application configuration parameters (such as codec or bit-rate) and q_f is the QoS the flow obtains. Examples of functional relations usable as $Q(\cdot)$ are given in Section 3.4 for video, voice and web QoE. We denote $q_{\text{req}}(f, p_f)$ the QoS so that $Q(f, p_f, q_{\text{req}}(f, p_f))$ is the highest. This is the requirement for a flow with parameters p_f to perform best. Finally, each flow f must be distributed on one single link,[3] $d_f = l \in L$, where $\mathbf{d}$ is the flow distribution vector, of size $\lvert F \rvert$.

The multihomed flow management problem thus consists selecting the network association for each interface ($\mathbf{a}$; possibly turning some off), distributing the flows over the active links ($\mathbf{d}$) and adjusting application parameters to the best matching set ($\mathbf{p}$). The concept of ABC terminals requires to maintain a high performance quality while keeping low power

[3] This work does not consider the possibility of multipath flows.

consumption and access prices. This triple objective can be expressed as

$$\max_{\mathbf{a},\mathbf{d},\mathbf{p}} \left(\sum_{f\in F} W_f Q\left(f, p_f, q_{\text{req}}(f, p_f)\right) \right. \\ \left. - W_b \sum_{i\in I} Pw(l_i) - W_p \sum_{i\in I} Pr(l_i) \right), \tag{3.9}$$

where the W_* are weighting factors which can be used to scale performance metrics to comparable ranges, and express their relative priority. Also, the following structural constraints apply,

$$\begin{cases} \forall f \in F, \exists i \in I \,/\, (a_i \neq \text{None}) \wedge (d_f = l_i), & \text{(3.10a)} \\ \forall i \in I, \quad \displaystyle\sum_{f\in F|d_f=l_i} C\left(q_{\text{req}}(f, p_f)\right) \leq C(QoS(l_i)). & \text{(3.10b)} \end{cases}$$

In other words, (3.10a) attributes a single active link to each flow and (3.10b) ensures that the maximal capacity available on each interface is respected.

A noteworthy fact about this model is that, in the process of optimising (3.9), it derives the QoS that the flow is expected to receive (for example its throughput). This information can be reported as a hint to the transport protocol the flow uses in order to skip its adaptation phases and directly adjust the rate to the selected conditions.

3.3.2 Comparison to QoS-based Decisions

To evaluate the potential improvements that could be achieved by the increased awareness of user and application requirements, we compare our proposal to two other mechanisms: 1) selecting only the network with the highest capacity and 2) load balancing flows over all interfaces, with each of the interfaces connected to the highest capacity uplink. The objective functions and additional required constraints for those two mechanisms are described below.

Network Selection

This first mechanism consists in selecting a single network which can provide the highest capacity. It is similar to what is currently done in consumer devices such as smart-phones (*e.g.*, Android-based or iPhones; Wasserman and Seite, 2011). These devices use a Wi-Fi connection in preference to the 3G connection. This default policy is logical as, in general, the Wi-Fi connection is likely to provide a higher capacity at a lower cost of network use.

The network selection problem can thus be represented as

$$\begin{aligned} &\max_{\mathbf{a}} \sum_{i\in I} C(l_i) \\ &\text{s.t.} \begin{cases} \exists i \in I \,/\, a_i \neq \text{None}, \\ \forall j \in I - \{i\}, \quad a_j = \text{None}. \end{cases} \end{aligned} \tag{3.11}$$

Flow Load Balancing

For the flow load balancing scheme, we consider a device which associates all its interfaces with their respective best networks, chosen by any or other QoS criteria. The current flows are then distributed so that each interface is equally loaded with respect to their available capacity (similar to Kandula *et al.*, 2008; Yao *et al.*, 2009).

To maintain loads roughly equal on all the interfaces of the terminal, we use a formula based on Jain *et al.* (1984)'s fairness as an additional optimisation objective. Rather than intrinsic capacity usage, we are interested in load balancing the flow requirements over links according to their capacity. We thus define for each link l a load ratio,

$$Lr(l) = \sum_{f \in F | d_f = l} \frac{C(q_{\text{req}}(p_f))}{C(l)},$$

which we use in the fairness index,

$$F_r = \frac{\left(\sum_{i \in I} Lr(l_i)\right)^2}{|I| \sum_{i \in I} Lr(l_i)^2}. \tag{3.12}$$

This index is 1 when all load ratios are equal, and tends to 0 with increasing unfairness in the load.

The load balancing problem can thus be formulated as

$$\max_{\mathbf{a},\mathbf{d}} \left(W_c \sum_{i \in I} C(l_i) + W_f F_r \right), \tag{3.13}$$

which represents the ideal load-balancing, probably better than what actual (sub-optimal) algorithms can achieve.

3.4 Using QoE Models for Network Selection

Not all applications serve the same objective. Therefore, there are different ways to measure the observed quality. Depending on the type of application, different quality metrics will be used. As presented in Section 2.4.2 (page 41), there has been a large body of research and standardisation work in the telecommunications community to provide QoE quality profiles. We reuse these models here to estimate the achievable QoE of interactive flows for various sets of application parameters and available access networks' QoS.

3.4.1 Motivational Example for Quality-Based Flow Management

Quality profiles tend to exhibit a non-linear behaviour. Figure 3.11 shows an example of such a behaviour for a video encoded with H.264 at different bit-rates, based on (2.5). Without knowledge of the application, reducing its allotted QoS can have adverse consequences on its quality. This strongly supports our hypothesis that application quality metrics are more relevant than raw QoS for flow management.

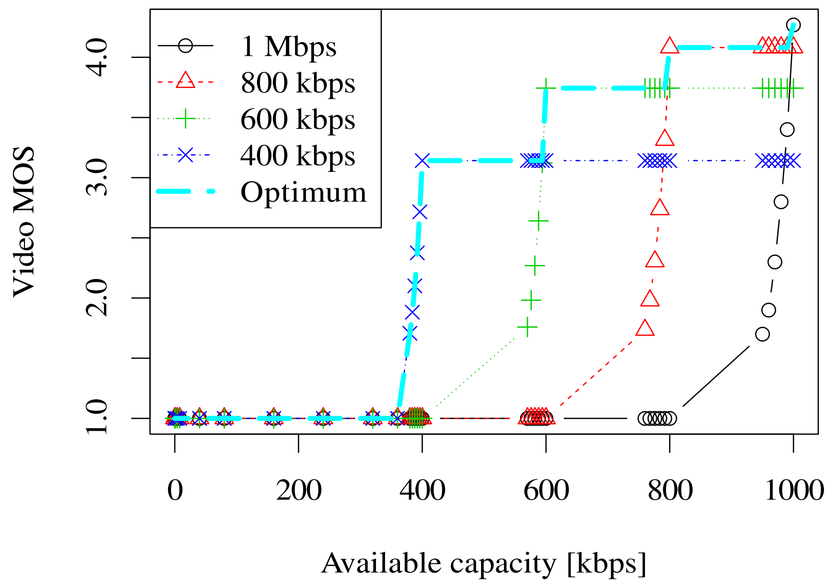


Figure 3.11 Quality profiles (2.5) for a video stream encoded with H.264 at various rates for varying network capacities. The optimum MOS is not a linear function of the available capacity. At a chosen bit-rate, a reduction in capacity may make no difference or may make the video unusable.

Application quality information allows more flexibility in the distribution of flows and allows to safely consider trade-offs to reduce power consumption and price of access while still providing an acceptable performance across all applications.

3.4.2 Assumptions for the Objective Models

Though we reuse the models presented in Section 2.4.2 (page 41), we make some assumptions about some of their parameters.

ITU models for real-time applications include information about packet error rate. However, our QoS measurements did not include it as they used TCP to probe the network path. Instead, we estimate this parameter based on the available capacity and the flow bit-rates. As discussed by Floyd and Fall (1999), it is not recommended not to use congestion control in the public Internet. Congestion-controlled transport protocols may however limit the throughput of an application to match the path's congestion level. For real-time applications, such limitation to a rate lower than their nominal bit-rate will incur packet losses as per (3.6). Therefore, if $C(q_f) < C(q_{\text{req}}(f, p_f))$, a ratio

$$plr = \frac{C\left(q_{\text{req}}(f, p_f)\right) - C(q_f)}{C\left(q_{\text{req}}(f, p_f)\right)}$$

of packets is lost. This ratio is used as the packet error rate component (e) of the QoS tuple to compute the performance quality of a flow with this reduced capacity as

$$Q\left(f, p_f, \langle c_f, r_f, 0, \ldots\rangle\right) = Q\left(f, p_f, \langle c_{\text{req}f}, r_f, plr, \ldots\rangle\right),$$

where $c_f = C(q_f)$, $r_f = R(q_f)$ and $c_{\mathrm{req}f} = C\left(q_{\mathrm{req}}(f, p_f)\right)$. This assumes that packets lost due to too small a capacity are not retransmitted. This is reasonable as the time it takes to retransmit a packet makes it useless for real-time media streaming.

The R factor (2.2) includes, more specifically for voice conversations, a parameter I_s which is a combination of simultaneous impairments not related to network conditions. However, there is no default or well-known value for this parameter. In the following, we follow the approach of Graubner *et al.* (2010) and ignore this term by setting it to 0.

Finally, the QoE profile for elastic traffic (2.6) considers a web session to start with a query through a web-based search engine or directory (T_1 and T_2 in $WeightedST$). We however note that not all web sessions start with a search query (*e.g.*, bookmarks or e-mails). We therefore propose to only consider the time needed to load the actual page by setting $T_1 = T_2 = 0$ and using

$$WeightedST = 0.98 \cdot T_3 + 1.76 \cdot T_4, \tag{3.14}$$

where $T_3 = r_f$ and $T_4 = s/c_f$. We will use (3.14) in the rest of this thesis.

3.5 Evaluation Scenarios

To evaluate and compare the performance of all three approaches, we implement them in the MiniZinc modelling language (Marriott *et al.*, 2008) and evaluate the various approaches in multiple scenarios. Each scenario differs in the number and type of interfaces, the possible links and their achievable QoS, and the number and type of flows. All parameters come from empirical measurements. This section first details our implementation of the model, then explains how evaluation scenarios were created.

3.5.1 MiniZinc Models

The models are relatively straightforward representations of the constraints expressed in Section 3.2. The functional relationships between network and monetary cost; network and interface, and QoS; and network and interface, and power consumption, are all represented as table constraints. Furthermore, the functional relationship between application parameters and QoS and QoE is also expressed as a table constraint. In this latter case the table is a discrete approximation of the functions. Consequently, the model has the form of a traditional constraints satisfaction problem (CSP; Dechter, 2003) for which we wish to optimise the objective function.

The MiniZinc language is supported by several different solvers. To run the scenarios we used the default solver. This solver employs the standard constraint programming approach: it uses constructive search, constraint propagation, and branch-and-bound pruning to completely explore the space of possible solutions and find the optimal one.

3.5.2 Numerical Parameters

The empirical data-sets collected and analysed in Section 3.2.3 have been used as the basis to create evaluation scenarios. They are summarised below, along with some assumptions which had to be made about them.

QoS The tuples of QoS parameters observed during the collection campaign described in Section 3.2.3 have been used to provide realistic network conditions.

Battery consumption We used the energy consumption data from Petander (2009). This data-set only covers 3G and Wi-Fi interfaces. As an equivalent for WiMAX adapters was not available, we have used the Wi-Fi measurements as nominal WiMAX values. We believe that this does not qualitatively change the outcome of our comparison.

Access Price The presented model does not encompass all different pricing methods which have been identified. However, as a first approximation, only pricing per connection time has been taken into account. Additionally, all Wi-Fi networks were considered cost-free.

Quality of experience The QoE for video, audio and web flows has been computed and tabulated for a wide range of supporting QoS, based on the models presented in Section 2.4.2 (page 41) and our assumptions from Section 3.4.2.

Web Demand Unlike real-time constant-bit-rate streaming, web traffic does not have a throughput requirement. Rather, as shown by (2.6), the perceived quality depends on the duration of the transfer which, in turn, depends on both the available capacity and the page size. We therefore used another data-set from Petander (2009) for the size distribution of web pages.

Priority and Scaling Factors Though the optimisation objective functions (3.9) and (3.13) include weighting factors W_* for all singular objectives, they are all equal and set to 1 for now. It is the subject of future work to study how to adjust them from user feedback.

3.5.3 Generic Scenarios

We first consider a set of generic scenarios which covers a wide range of use-cases for mobile devices (*e.g.*, smart-phones or vehicular routers). Those scenarios are synthetically generated by a random process. The numbers of interfaces, access networks and flows are first determined. Then, the power consumption, cost and QoS of the links are randomly chosen from the data-sets. The identifier of the scenario is used as the seed of the pseudo-random number generator in order to allow identical recreation. The parameter ranges for the scenario generation are shown in Table 3.3.

Figure 3.12 shows the parameters of the scenarios we considered. We arbitrarily chose the first 100, though 5 of them did not have any available networks, reducing the number of significant scenarios to 95. For each scenario, the three approaches have been evaluated by

Table 3.3 Parameter ranges used for the generation of the synthetic scenarios.

Technology	**Interfaces**		**Networks**	
	Min	**Max**	**Min**	**Max**
3G	1	1	0	1
Wi-Fi	1	2	0	10
WiMAX	0	1	0	2

Flow type	**Min**	**Max**
voice over IP (VoIP)	0	3
Video	0	3
Web	5	10

the constraint solver to find the optimal solution each technique could yield. In order to study how the different approaches perform under different demands, the number of flows has been varied from 1 to the total number for each scenario. This allowed us to observe the variation of overall performance metrics, which we discuss in the next section.

3.5.4 Typical Smart-phone Use Scenarios

We additionally study the performance of the proposal for more specific scenarios of the smart-phone use-case. We consider a two-way video conversation and web browsing. This gives a fixed demand of 2 VoIP flows, 2 video flows, and 3 web sessions. The sizes of the web sessions are taken from the nearest rank from the 40, 50 and 60^{th} percentiles of the data-set. We also limit the scenarios from the previous sub-section to those which have a single Wi-Fi interface, as is currently the case for hand-held mobile devices. The available networks remain the same.

3.6 Results and Discussion

We ran the scenarios presented above for our combined approach as well as for the network selection and load balancing schemes. The number of flows from each scenario was varied from 1 to the maximum of each scenario to study the behaviour of the different approaches under various demands. However, the solving models were not implemented with speed of solving in mind. As a result, the time to find the optimal solution increases more than linearly with the number of flows.

To avoid unmanageable completion-waiting times, we added a stopping condition in the iteration from 1 to $|F|$ flows if the most recent solution took more than a given time to find.[4] This stopping criterion has the adverse effect of limiting the number of samples for large numbers of active flows, which widens the confidence interval of the results. However, with fewer than 7 concurrent flows, solving the problem often takes less than 20 s.

[4]The scenarios were evaluated on a cluster made of 2 GHz Xeon machines running Redhat Linux with kernel 2.6.18-92.1.13.el5 #1 SMP. However, the solver did not make use of the multiple cores of the machine, and ran in parallel with other jobs.

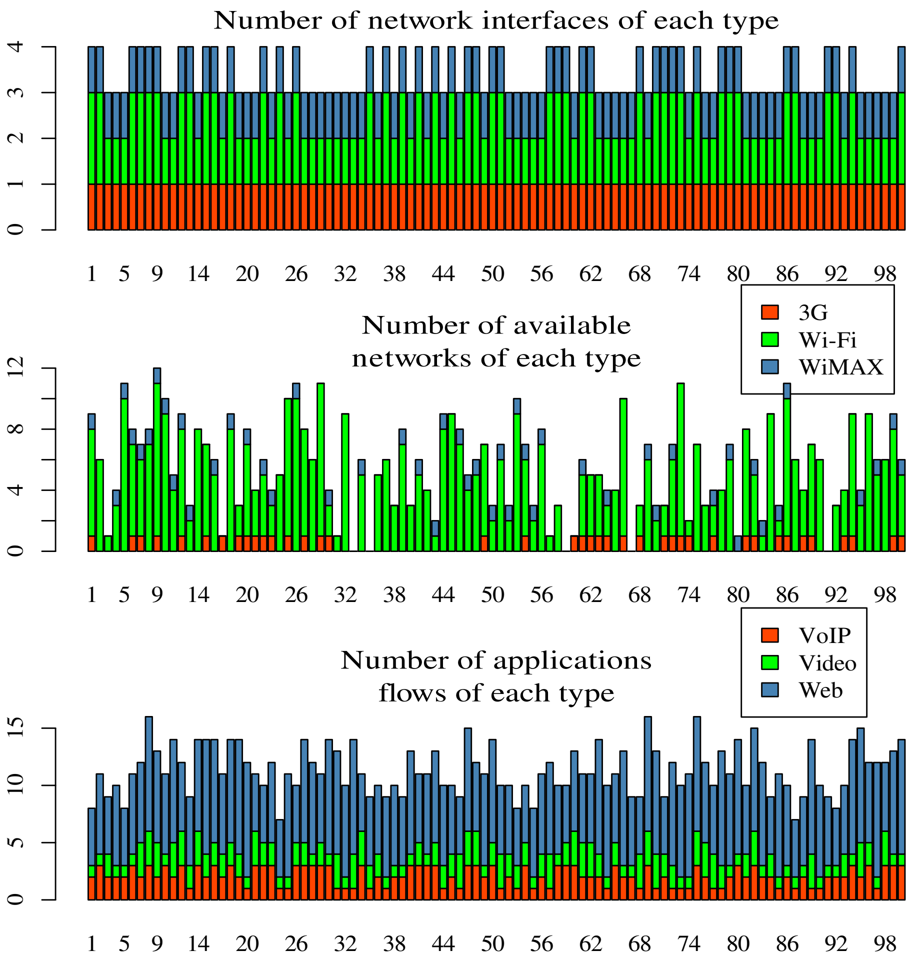


Figure 3.12 Scenarios parameters. Top: number of interfaces and available networks of each technology in the studied synthetic scenarios; Bottom: number of flows per type.

For all scenarios, we focus on the average quality over all flows, as well as the power consumption and the price. The latter two metrics have been transposed to units more meaningful for a user: consumption of the full battery in %/s, and price in ¢/s.

3.6.1 Generic Scenarios

Figures 3.13, 3.14 and 3.15 respectively compare the average quality, power consumption and access price achieved depending on the approach. Both overall averages and median values are reported. For the average plots, error bars show the standard error of the results; for the median ones, they are placed at $\pm 1.58 \cdot IQR/\sqrt{n_s}$, where n_s is the number of samples for the given number of flows, as an estimate of the 95 % confidence interval. Only values for which $n_s > 20$ experiments finished within the deadline are plotted.

Figure 3.13 shows the variation in the achievable QoE. As could have been expected, the network selection scheme quickly delivers bad quality because it tries to fit all flows over a single link with limited capacity. The load-balancing approach performs better here, provided there is more than one link available for distributing the flows. The QoE-aware decision system, however, manages to maintain the average quality consistently between 4 and 5. It even increases with a larger number of flows, as it becomes worth enabling more interfaces to better support the demand.

The power consumption is shown on Figure 3.14. The network selection scheme, using only one interface at a time, usually has the lowest battery consumption. The load balancing, which uses all its interfaces, regardless of the needs, always uses a larger amount of battery. Our proposal has a more dynamic power consumption, which increases as it establishes more links to cater for a higher demand.

Finally, the price is shown on Figure 3.15. In the same way as the power consumption, it is directly related to the number of established links. As all the Wi-Fi networks, with the highest capacity, were considered public and free in our scenarios, the network selection scheme unsurprisingly yields a rather low price. The load balancing approach establishes links even on for-a-fee networks, and tries to distribute traffic evenly on them. Using this technique therefore results in higher prices overall. As one of the objectives of our QoE-aware proposal is to keep the price to a minimum, it rarely uses costly network where alternatives exist, even if with lower QoSs, as it can adapt application parameters accordingly.

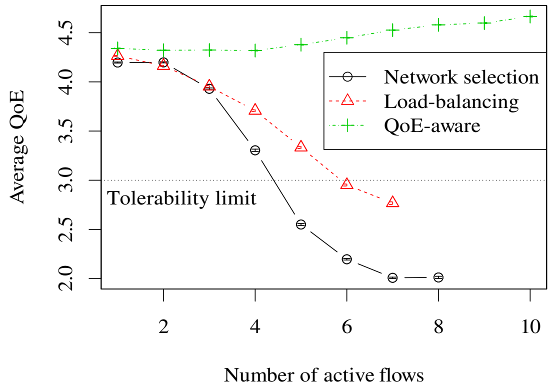
3.6.2 Smart-phone Scenarios

Figure 3.16 compares the performance of the proposed performance metrics-aware multihomed flow management technique to the network selection and load balancing approaches with a fixed realistic demand over the subset of scenarios (56) which include a single Wi-Fi interface. We note that the single network selection approach currently implemented in smart-phones provides, on the average, the worst QoE performance and that our proposal, while maintaining a high QoE, manages to keep the median battery consumption in between those of the two others and keep the access price to a minimum.

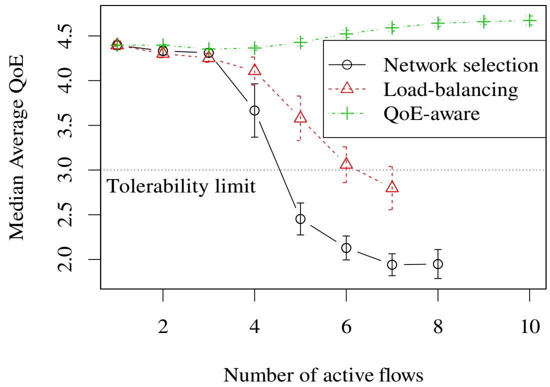
From those two sets of results, it appears that awareness of application parameters and QoE metrics allows to make better decisions with respect to which network to connect to, and how to distribute application flows on them. The possibility to manipulate application parameters based on the knowledge of the QoS they will encounter permits keeping the overall perceived experience high, while maintaining low battery consumption and access costs.

3.7 Conclusion and Future Work

In this chapter, we have argued that QoS-based network selection is not sufficient to provide a good experience to the user. We introduced a user- and application-aware decision mechanism for multihomed mobile devices to support this. We devised it to select access networks to use for each interface, distribute the application flows over those links and configure the applications to maintain a high quality of experience and keep

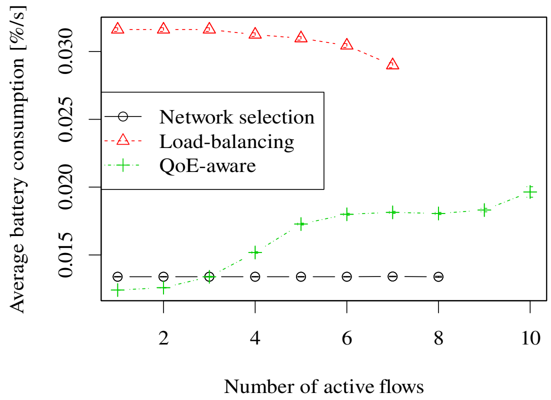


(a) Averages

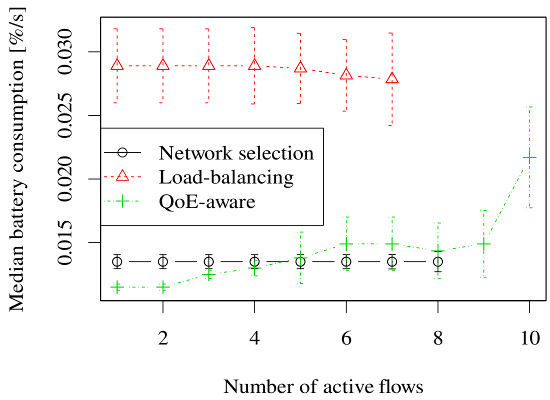


(b) Medians

Figure 3.13 Average QoE achieved by the three decision schemes. The QoE-aware approach outperforms the two others as it is the only one to manipulate application parameters to match the current QoS.

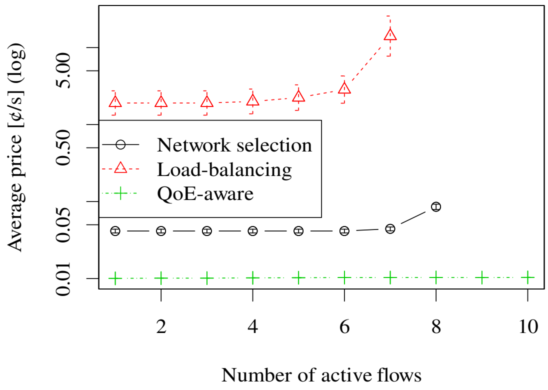


(a) Averages

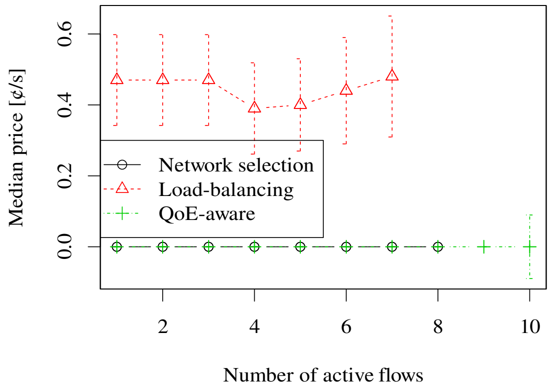


(b) Medians

Figure 3.14 Battery consumption depending on the number of active flows for each approach. A consumption of 0.03%/s results in a battery lifetime of less than an hour. The apparent decrease in consumption of the load-balancing strategy is biased by the limited number of scenarios with 8 or more flows which could run without hitting the time limit.



(a) Averages



(b) Medians

Figure 3.15 Price per second of the solutions found by the presented approaches. Only the QoE-aware scheme is mindful of that metric, and tries to keep it as low as possible by using free networks.

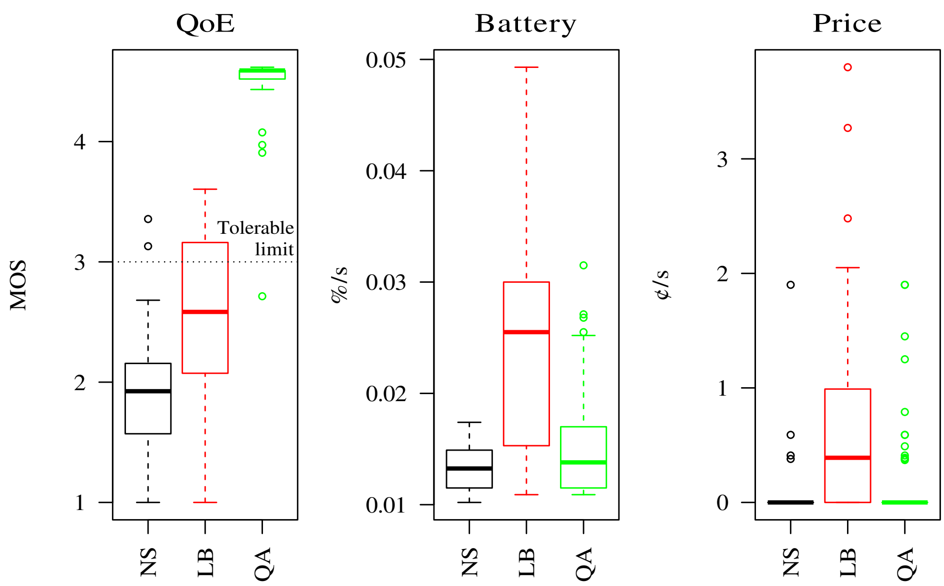


Figure 3.16 Distribution of the overall performance metrics yielded by the QoE-aware multihomed flow management (QA), network selection (NS) and load-balancing (LB) approaches in the static demand evaluation. Our approach outperforms the two others in terms of application quality, while keeping reasonably low power consumption and price.

other perfunctory factors such as battery consumption and access price as low as possible. Performance indices other than QoE can also be used for non-interactive applications.

To evaluate our proposal, we modelled it as a COP. We compared its performance in terms of overall performance to two common decision mechanisms: the single network selection approach which is currently used in most smart-phones, and multihomed load balancing. Experimental results have shown that the proposed quality-aware approach out-performs others by supporting high application quality while keeping power consumption and price to a minimum. We also showed how those results remain valid with increasing demand.

The results presented in this chapter are encouraging and lead to interesting future work. First, some assumptions of the model should be removed to get more accurate quantitative results.

- The direction of flows was not considered (all were assumed to be downloads with respect to link capacities); networks' asymmetric QoSs should be taken into account in future iterations of the model.
- The price modelling does not properly reflect the actual complexity of current operators' practice;
- WiMAX power traces were missing, and field experiments should be conducted to acquire those;
- Encapsulation and tunnelling overhead (*e.g.*, going through the HA) was not accounted for;
- Due to the way QoS measurements have been performed, they were not as accurately decomposed as our initial model was; specific work on separating the contribution of the various components of a network path would benefit our approach as much finer decisions could be taken.
- It was assumed that the decision mechanism was given full information and full control on the flow it had to manage; this is acceptable in the case of an MN originating all the flows but, in the case of a mobile router merely managing flows on behalf of mobile network nodes, information and control systems for NEMOs have to be studied (see Ben Rayana and Bonnin, 2008; Gaultier *et al.*, 2009, for example).
- The presented optimisation method considered the operation of a single device. Study of the stability of a system where all users are using such a mechanism has been left out. However, synchronisation problems and oscillatory patterns could emerge from a global use of our proposed decision system, such as users joining and leaving the same networks simultaneously. This should be considered, and methods to avoid it should be studied and included in further studies on this decision algorithm and its criteria.

Second, though weights were introduced in the objective functions, they were not used. How to let the user give feedback to the system to adjust those weights and provide solutions more finely tailored to their needs should be explored. Possible approaches include using Likert-type scales (Likert, 1932) to let the user input their opinion in real-time and use techniques such as the analytic hierarchy process (Saaty, 2000) to adjust the weights accordingly.

Finally, as was discussed when presenting the results, the CSP-based implementation was not optimised for solving time. It is important that the mechanism is able to run in real-time, either on a mobile device or delegated to a decision "provider," to benefit from optimal QoE-aware decisions in a real system. Work on improving the CSP representation, comparing the performance of various solvers, and considering restarting the optimisation from partial or previous solutions is therefore important. Alternatively, other approaches such as linear programming techniques could be used to search the solution space.

Chapter 4

Mobility-Aware Transport Protocol

4.1 Introduction

Following the increased connectivity and ease of use of constantly connected mobile devices, there is also a shift towards real-time applications such as multimedia streaming, voice over IP (VoIP) or video conferencing in mobile environments. However, the supporting protocols to transport the real-time traffic of emerging applications are still mainly those designed to carry best effort traffic for fixed and wire-connected devices, which were reviewed in Section 2.1.1 (page 13).

While UDP has historically been used to carry real-time traffic, it does not provide congestion control. Such a feature is however highly desirable in a shared network infrastructure (Floyd and Fall, 1999). The Datagram Congestion Control Protocol (DCCP), proposed by Kohler *et al.* (2006a,b), is a non-reliable but congestion-aware transport protocol which could efficiently replace UDP for such purpose. However, as described in Section 2.1.2 (page 17), the performance of congestion-controlled transport protocols suffers in mobile wireless environments due to several underlying assumptions being broken. DCCP's congestion control mechanisms are not immune to these issues.

In this chapter, we therefore propose a mobility-aware enhancement to one of DCCP's congestion control mechanisms, the TCP-Friendly Rate Control (TFRC; Floyd *et al.*, 2000; Widmer, 2003; Floyd *et al.*, 2008). We consider handovers between networks with potentially heterogeneous characteristics. In our generic scenario, depicted in Figure 4.1, a wireless mobile node (MN) moves between two or more networks while having established sessions with fixed correspondent nodes (CNs) in the Internet. The handover can be between two access points of the same technology, that is, horizontal, or vertical, between heterogeneous wireless networks such as 3G to Wi-Fi.

We first study the behaviour of standard DCCP/TFRC in mobility situations, and numerically model the effects of such handovers on TFRC. We use this model to quantit-

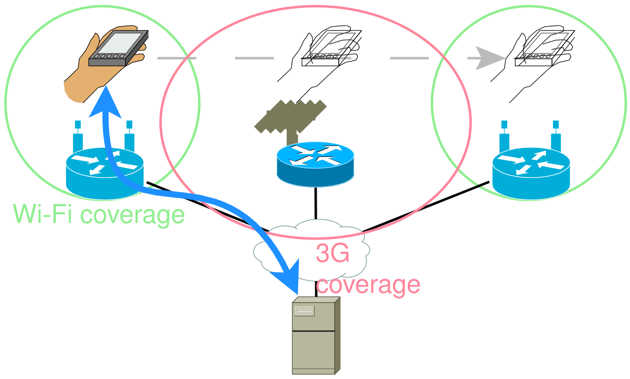


Figure 4.1 Generic use-case scenario: vertical handovers between a number of access networks with different characteristics.

atively derive the potential for improvement. We then present an end-to-end protocol extension designed to better cope with mobility events and adapt faster to different quality of service (QoS) on the new access network. Our proposal relies on explicit notifications—similarly to, *e.g.*, the work of Montavont and Noël (2006)—and can be directly integrated as an enforcement element in our cross-layer framework (Figure 4.2). This disconnection-tolerant modification of DCCP/TFRC shares similarities in concept with Freeze-TCP (Goff *et al.*, 2000) but has different target applications and introduces further enhancements.

This chapter is structured as follows: Section 4.2 uses simulations to highlight the impact of mobility-induced disconnections on TFRC and introduces an numerical study of its behaviour, allowing to estimate possible performance gains; Section 4.3 presents the proposed TFRC protocol modifications and their implementation into Freeze-DCCP/TFRC; simulations and experimental results with a Linux implementation are presented in section 4.4. Finally, in Section 4.5, we summarise this work and present future research directions.

4.2 Behaviour of TFRC Over a Disconnection

In this section, we investigate the issues TFRC faces when used in mobile scenarios with heterogeneous handovers. We first describe the operation of TFRC as standardised by Floyd *et al.* (2008). We then provide an example simulation highlighting some of the issues, before modelling the behaviour of this rate control to quantify the performance discrepancies.

4.2.1 Operation of Standard TFRC

Based on feedback from the receiver, a standard TFRC sender controls its rate X following a model of Transmission Control Protocol (TCP)'s throughput under the same

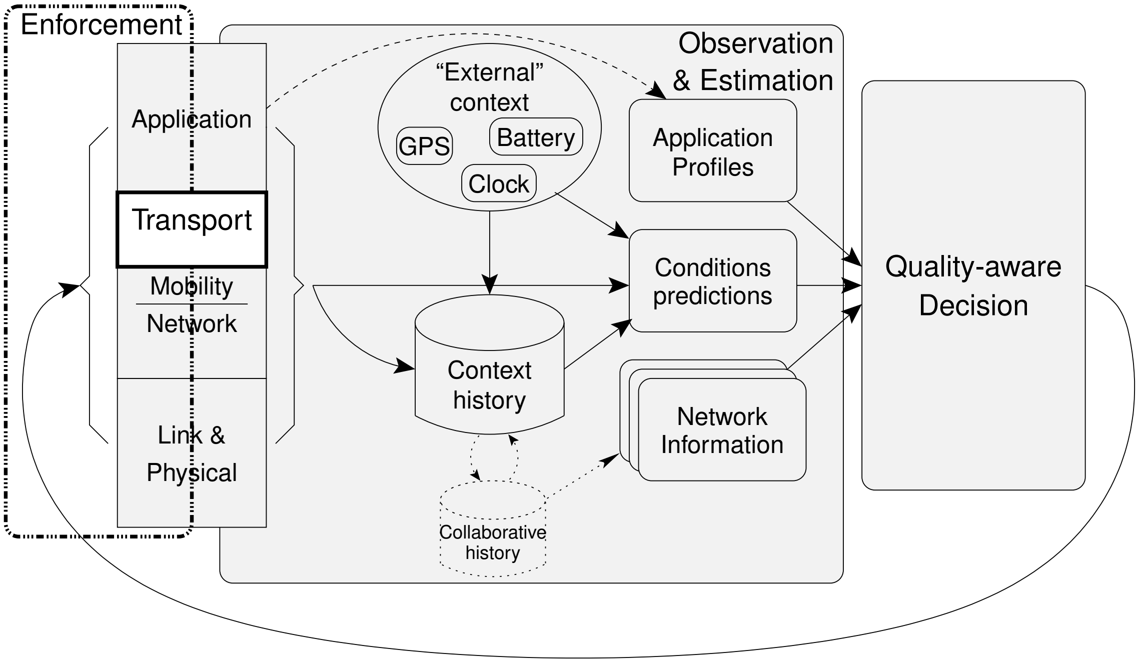


Figure 4.2 Relation of the contribution presented in this chapter to the cross-layer framework of this thesis. Bold lines and non greyed-out components are the current focus.

conditions (Padhye *et al.*, 1998),

$$X_{\mathrm{Bps}} = T(p,R) = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3p}{8}}\right)p\left(1+32p^2\right)}, \tag{4.1}$$

$$X \leftarrow \min(X_{\mathrm{Bps}}, 2X_{\mathrm{recv}}) \tag{4.2}$$

where s is the packet size, R the round-trip time (RTT), and t_{RTO} the retransmit timeout (usually $4R$). Parameters p and X_{recv} are reported by the receiver roughly every RTT and are, respectively, the loss event rate, and the current received rate. If no report from the receiver is seen before t_{RTO} expires, the sender reduces its allowed sending rate by halving the last used X_{recv}.

As for TCP, a slow-start phase is also present to first adapt the rate to the network path's capacity. During this phase, the sender updates its rate once per RTT following

$$X \leftarrow \min(2X, 2X_{\mathrm{recv}}), \tag{4.3}$$

until the first loss is observed. When the first loss occurs, the TFRC receiver reports a loss event rate p which reflects its observed throughput X_{recv} before the loss. The value of p is initialised by inverting (4.1).[1]

When further losses are observed, the TFRC receiver computes p as the inverse of the weighted average of the lengths of the n most recent loss intervals $i_0, \ldots, i_{n-1}$. The length

[1] Floyd *et al.* (2008) do not specify how this inversion should be done. Most implementations use a binary search, but Jourjon *et al.* (2007) suggest a more CPU efficient method based on a Newtown search.

of a loss interval is measured in number of packets successfully received. The average is computed using a vector of decreasing weights $\mathbf{w} = [w_0, \ldots, w_{n-1}]$ as $S_0 = \sum_{i=0}^{n-1} w_i i_i$. The TFRC receiver actually keeps a history $\mathbf{i} = [i_0, \ldots, i_n]$ of the last $n+1$ loss intervals. This slightly larger buffer is designed to avoid overly increasing the loss event rate when one of these events has just happened. Indeed, when losses have just been experienced, the size of the current loss interval i_0 starts increasing from 0. At first, i_0 is so small that it would incorrectly drive p up and needlessly reduce the rate $X_{\text{Bps}} = T(p, R)$. It is therefore ignored and the reported loss event rate is still based on the previous $i_1, \ldots, i_n$ intervals. As these values do not change anymore, p is stationary during this period. Taking $S_1 = \sum_{i=0}^{n-1} w_i i_{i+1}$, Floyd *et al.* (2008) therefore compute p as

$$p = \frac{1}{i_{\text{mean}}} = \frac{\sum_{i=0}^{n-1} w_i}{\max(S_0, S_1)}. \tag{4.4}$$

Computing p this way only considers the duration of loss-less periods, and is not related to the duration of periods during which all packets are lost, even if they span several RTTs.

As $T(p, R)$ has an inverse relation with the loss event rate p, TFRC is not fit to work on networks with loss-inducing disconnections. A temporary break in the end-to-end path would indeed have several consequences. First, packets will needlessly use parts of the network path's capacity before being dropped, resulting in losses. The sending rate will then gradually be reduced. Upon reconnection, the transport protocol's rate will therefore not match the network's characteristics and need some time to re-adapt. We illustrate this behaviour in the next section.

Moreover, as the computation of the loss event rate is based on a history of several loss events, TFRC reacts slowly to immediate decreases in p. In the case of a cross-technology hand-offs to access networks with a larger capacity, it will therefore take a much longer time to adjust the rate to the newly available capacity.

4.2.2 Simulation of Mobile Handovers

To present an example of the adverse consequences of disconnections on the sending rate of DCCP/TFRC in mobility situations, several simulations were run with *ns*-2 (ns-2 manual).[2] The simulation scenario consists of a landscape of 800×1600 m where a Mobile IPv6 (MIPv6) MN moves between the coverage of three access routers (ARs). Figure 4.3 presents the simulated environment, consisting of one backbone router and the three ARs providing non-overlapping wireless connectivity to the MN receiving traffic. Some simulations were also run with the second base station disabled in order to observe the behaviour of the data stream in the case of a more sporadically available network coverage.

ns-2 was configured to simulate a regular single-rate 11 Mbps 802.11b wireless channel. Some parameters had to be adjusted to obtain the desired simulation conditions: the wireless reception threshold has been fine-tuned to simulate a 400 m Wi-Fi range. Table 4.1

[2] A complete rewrite of this tool, ns-3, was available at the time of this work, but some of the main protocols were not yet implemented. This is no longer the case, and we believe it is time to move forward, and that it would now be ill-advised to keep using *ns*-2 for such simulations as the new version is much more flexible and has a cleaner and more extensible code-base.

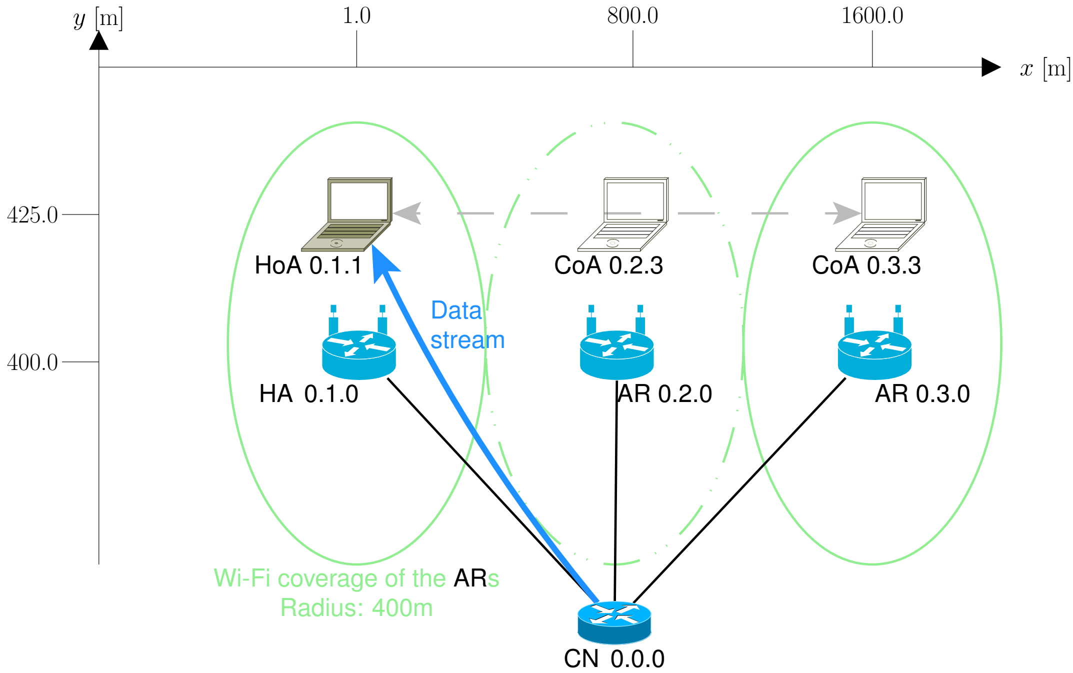


Figure 4.3 The basic simulation scenario. An MN moves back and forth between three adjacent (but not overlapping) wireless networks with different prefixes. The CN sends a constant stream of data to the MN using DCCP/TFRC. For simplicity, the MN's home agent (HA) is set to be the AR of the first access network. Addresses are expressed in *ns*-2 format.

summarises these configuration changes. We have also ported the MobiWan[3] (Ernst, 2001) MIPv6 support and DCCP module[4] (Mattsson, 2004) to version 2.33 of this simulator, and updated these to the latest versions (at the time) of their respective specifications (Johnson *et al.*, 2004; Floyd *et al.*, 2008).[5] All kinds of route optimisations for MIPv6 were disabled.

In the first scenario, the MN moved back and forth at constant speed between all three ARs (from adjacent Wi-Fi networks), losing connectivity with the current one, associating with the new one, and re-establishing its mobility bindings with its HA. Figure 4.4 shows that, in addition to the delay to associate with the new AR and re-establish the bindings when the previous link breaks, there is a delay between the time when a CoA is fully configured on the new access network, and when the rate of TFRC is reinstated: 100 ms until it restarts, but 500 ms until it is fully restored. Figure 4.5 shows the results for a similar scenario where the second access point has been disabled, thus creating a period of complete lack of connectivity. The previous delay effect becomes much larger, up to 50 s, in this case.

In the next section, we model this behaviour in order to evaluate the performance issues in terms of lost packets, delay until restart and "wasted" capacity.

[3] http://www.inrialpes.fr/planete/mobiwan/
[4] http://lifc.univ-fcomte.fr/~dedu/ns2/
[5] These updated patch-sets are available at http://www.nicta.com.au/people/mehanio/nsmisc/.

Table 4.1 Parameters adjusted from *ns*-2's defaults.

***ns*-2 parameter**	**Value**
11 Mbps 802.11b channel	
`Phy/WirelessPhy bandwidth_`	`11Mb`
`Phy/WirelessPhy freq_`	`2.472e9`
`Mac/802_11 dataRate_`	`11Mb`
`Mac/802_11 basicRate_`	`1Mb`
Miscellaneous	
`Phy/WirelessPhy RXThresh_`	`5.57346e-11`
`Agent/MIPv6/MN bs_forwarding_`	`0`
`Agent/MIPv6/MN rt_opti_`	`0`

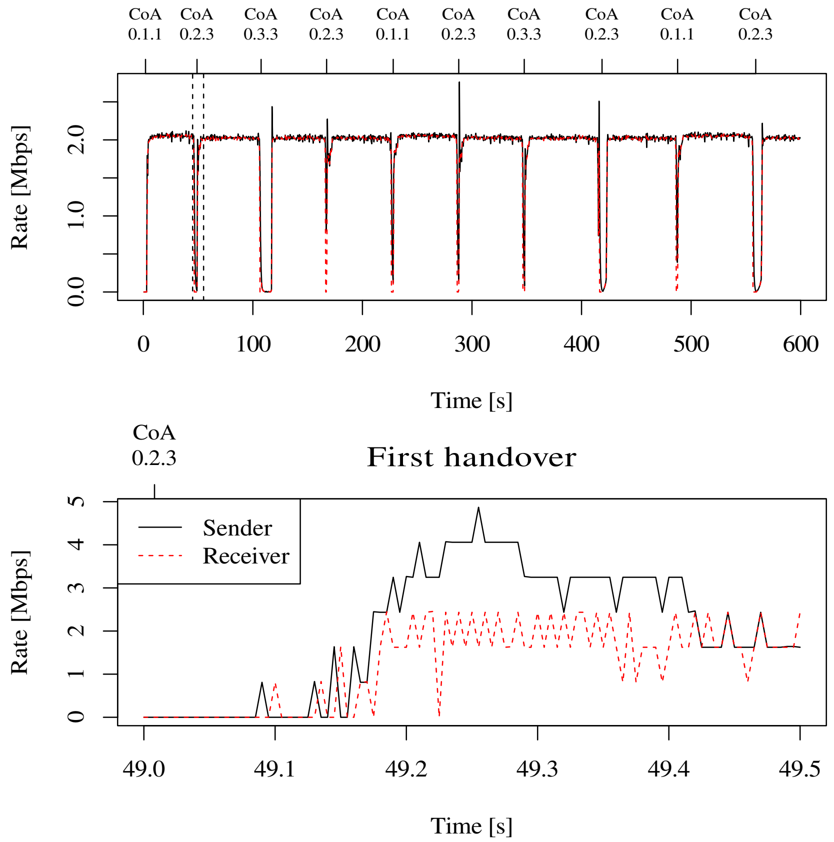


Figure 4.4 DCCP/TFRC data stream moving through adjacent Wi-Fi access networks. Labels on the top axis represent the time when the new CoA has been configured and is fully usable.

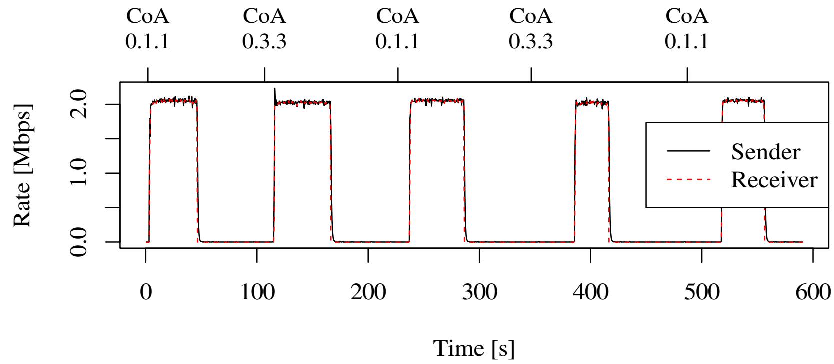


Figure 4.5 DCCP/TFRC data stream moving through non-adjacent Wi-Fi access networks.

4.2.3 Numerical Model of TFRC's Behaviour

In order to quantify the highlighted impact, we introduce a model of TFRC's behaviour when a disconnection occurs. It is used to derive the number of packets that are lost during the disconnection, the delay before TFRC resumes sending after a reconnection, the available capacity and the time it takes to adapt to the new characteristics. After validating it with *ns*-2 simulations, we evaluate these metrics for various typical horizontal and vertical handover scenarios. This shows that there is ample room for better management of these events. A summary of the symbols used throughout this section is given in Table 4.2.

During the Disconnection

Evolution of Internal Parameters We evaluate the changes in sender rate X during a disconnection, as well as the `nofeedback` timer period t_{RTO}. Both values have an impact on the number of packets lost during the disconnection and the rate recovery after the reconnection. Figure 4.6 represents the evolution of these parameters.

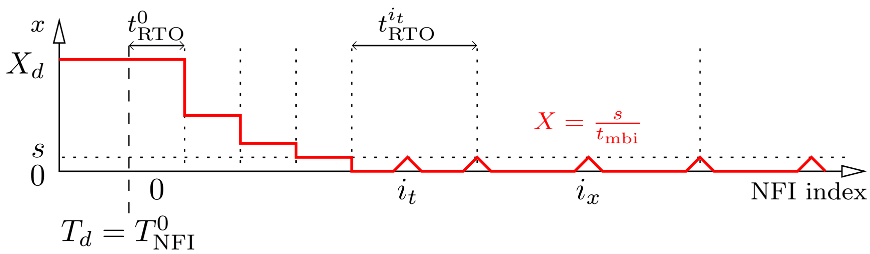


Figure 4.6 The evolution of TFRC's internal parameters after a disconnection.

Just before the disconnection occurs, at T_d, the sender sends at rate $X = X_d$. In the following, this is assumed to be the nominal TFRC rate that the underlying link can support. Consequently, the receiver measures and reports an X_{recv} roughly equal to X_d. Therefore, (4.2) is limited by X_{Bps}.

Table 4.2 Notations used for the analysis of TFRC over a disconnection.

Symbol	Meaning
	During disconnection
t^i_{RTO}	Duration of no feedback interval (NFI) i
X^i	Sender rate during NFI i ($X_d = X^0$)
i_x	First NFI so that $X^{i_x} = s/t_{\mathrm{mbi}}$
i_t	First NFI so that $t^{i_t}_{\mathrm{RTO}}$ starts increasing
n_{lost}	Number of packets lost during the disconnection
	After reconnection
p_r	Loss event rate upon reconnection, entirely based on the previous network
$X_{\max}$	Maximal TFRC sender rate the new network can support
t_{idle}	Time before the first packet is sent after reconnection
X^i_r	Sender rate during RTT i after the reconnection ($X_c = X^0_r$)
R^i_r	Sender's estimation of the RTT of the new network
n^{ε}_R	Number of RTTs on the new network before R^i_r is within ε of R_{new}
n^i_{pkts}	Total number of packets sent after RTT i on the new network
$t_{\mathrm{ss,recov,grow}}$	Times to recover from the disconnection and adapt to the new capacity
n^*_{wasted}	Number of packets that could have been sent

In the absence of feedback, the TFRC sender halves its allowed sending rate every time the `nofeedback` timer expires by reducing its local estimate of X_{recv}. When X becomes small, t_{RTO} is increased to cover the transmission of at least two packets.

For convenience, we segment the disconnected period into *no feedback intervals* (NFI). An NFI is the interval between two consecutive expirations of the `nofeedback` timer.[6] NFIs are indexed starting at $i = 0$. The first expiration of the `nofeedback` timer marks the end of NFI 0. Hence, the effects of this timeout start at the beginning of NFI 1. The rate then gradually decreases until it reaches its minimum value during NFI i_x.

Every NFI, the sender halves the value of X_{recv}, which then drives (4.2). In the worst situation, X can reduce to the minimal value of one packet every 64 seconds (s/t_{mbi}). Taking i_x as the NFI during which $2X^{i_x}_{\mathrm{recv}}$ drops below s/t_{mbi}, the sender rate can be expressed as

$$X^i = \begin{cases} \frac{X_d}{2^i} & \text{if } 0 \le i < i_x, \\ \frac{s}{t_{\mathrm{mbi}}} & \text{otherwise,} \end{cases} \tag{4.5}$$

$$i_x = \left\lceil \log_2 \frac{X_d \cdot t_{\mathrm{mbi}}}{s} \right\rceil, \tag{4.6}$$

where $\lceil\cdot\rceil$ is the ceiling operator.

Additionally, the `nofeedback` timer, initially set to $t^0_{\mathrm{RTO}} = 4R$, increases when the sending rate becomes smaller than $2s/4R$. Assuming $X_d \ge 2s/4R$ and taking i_t as the NFI during

[6] An NFI is the same concept as the NFT of Kohler *et al.* (2008).

which $2s/X^{i_t}$ becomes larger than $4R$, the duration of NFI i is then

$$t^i_{\mathrm{RTO}} = \begin{cases} 4R & \text{if } i < i_t, \\ \frac{2s}{X^i} & \text{otherwise,} \end{cases} \tag{4.7}$$

$$i_t = \left\lceil \log_2 \frac{2R \cdot X_d}{s} \right\rceil . \tag{4.8}$$

Note that (4.8) is only valid for $R < t_{\mathrm{mbi}}/2$, in which case $i_t \leq i_x$. Otherwise, $4R$ is larger than the time to send 2 packets at the lowest rate, and i_t is considered to be $+\infty$.

Packet Losses Though the number of losses happening during the single loss event of the handover does not directly impact TFRC's sender rate, they are an unnecessary charge on the rest of the network which could be better used for other traffic for which data can actually be delivered to the destination. It is interesting to quantify this charge on the network in the form of the number of packets which will eventually be lost during the time of the disconnection.

Figure 4.7 shows the evolution of the sender rate over a handover. Two cases are represented, for different reconnection times T_c and T'_c. They respectively occur before and after the sender's estimation of the receiver rate has reduced to less than one packet per RTT.

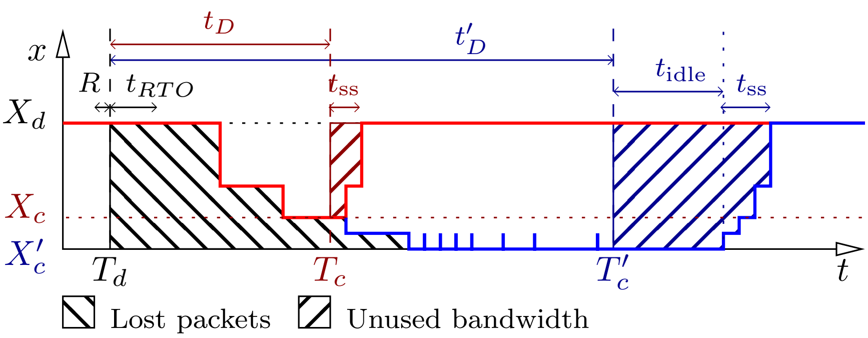


Figure 4.7 TFRC rate behaviour in a period with no connectivity. Two cases are shown, with different times of reconnection: at T_c, a time t_D has elapsed which is short enough that TFRC's rate didn't reach its minimum (red) and at T'_c, when the time t'_D elapsed since the disconnection was sufficient for X to reduce to s/t_{mbi}, an additional delay t_{idle} is present in this case before TFRC starts restoring its rate (blue).

Time $t_D = T_c - T_d$ is the length of the disconnected period. All the packets sent during this period are lost. The number of lost packets when the reconnection occurs, after n_D NFIs (such that $\sum_{i=0}^{n_D} t^i_{\mathrm{RTO}} \geq t_D$), can be estimated using

$$n_{\mathrm{lost}} = \begin{cases} \left\lfloor \frac{t_D X^0}{s} \right\rfloor & \text{if } t_D \leq t^0_{\mathrm{RTO}}, \\ \left\lfloor \frac{t^0_{\mathrm{RTO}} X^0}{s} + \sum_{i=1}^{i_D} \frac{t^i_{\mathrm{RTO}} X^i}{s} \right\rfloor & \text{otherwise,} \end{cases} \tag{4.9}$$

where $i_D = n_D - 1$ is the index of the n_D^{th} NFI and $\lfloor \cdot \rfloor$ is the floor operator.

After the Reconnection

Variation of the Loss Event Rate The losses will only be noticed by the receiver after reconnecting. Floyd *et al.* (2008) specify that the expected arrival time of a lost packet is interpolated using those of both packets received directly before and after the loss. Multiple losses over the disconnected period will then be considered part of the same loss event starting in the middle of the disconnected period.

Following the procedure described in Section 4.2.1, the evolution of p can be in three different phases:

no loss when $S_0 \geq S_1$, p gradually decreases as the number of received packets, in i_0, increases;

first loss observed makes $S_0 < S_1$ which stabilises p until the current loss interval i_0 becomes large enough so the inequality is reversed;

new loss observed when $S_0 < S_1$, $\mathbf{i}$ gets shifted which increases p as (4.4) now has a smaller denominator.

In the congestion avoidance state, the variation of p can be in different ranges depending which of the three phases it is in.

$$\begin{cases} \Delta p = 0 & \text{after the first loss has been observed,} \\ \Delta p > 0 & \text{when more losses are observed,} \\ \Delta p_{\min}(\Delta n_{\text{pkts}}, p_{\text{prev}}) \leq \Delta p < 0 & \text{otherwise (no loss).} \end{cases} \tag{4.10}$$

The lower bound of $\Delta p = p - p_{\text{prev}}$ when no losses occur can be derived using (4.4) to estimate p. If all Δn_{pkts} packets sent since the last feedback, which reported p_{prev}, have been received and taken into account in the receiver's next feedback the reduction in p will be

$$\Delta p_{\min}(\Delta n_{\text{pkts}}, p_{\text{prev}}) = \underbrace{\frac{\sum_{i=0}^{n-1} w_i}{w_0 \Delta n_{\text{pkts}} + (\sum_{i=0}^{n-1} w_i)/p_{\text{prev}}}}_{p=1/i_{\text{mean}}} - p_{\text{prev}}. \tag{4.11}$$

Event Timing While the time base of changes of the disconnected sender are regulated by the length of its retransmit timeout, feedback from the receiver takes on this role once reconnected. The receiver sends periodic feedback messages at least once per RTT. For the rest of this section, time will then be segmented in units of RTTs. Indices i now refer to how many of those have elapsed since the reconnection, rather than NFIs as previously.

The periodicity of sender-side events triggered by these feedback messages will follow the RTT of the visited network. While this value is mostly stationary for a given network, the sender does not use it directly for its computations, most notably that of the sending rate. To ensure a smooth evolution, it uses an exponentially weighted moving average of the past samples to estimate the RTT. After feedback message i allowing to sample RTT R_{new} of the new network, the sender's estimate of the RTT is

$$R_r^i = qR_r^{i-1} + (1-q)R_{\text{new}}, \tag{4.12}$$

with $0 < q < 1$ ($q = 0.9$ in Floyd *et al.*, 2008). Just after the reconnection, before the first feedback message has been received, the estimate completely reflects the RTT of the previous network, $R_r^0 = R_{\text{old}}$. Expanding the series, this estimate can be expressed as

$$\begin{aligned} R_r^i &= (1-q)R_{\text{new}} \underbrace{\sum_{j=0}^{i-1} q^j}_{\frac{q^0-q^i}{1-q}} + q^i R_r^0 \\ &= \left(1-q^i\right) R_{\text{new}} + q^i R_{\text{old}}. \end{aligned} \tag{4.13}$$

It will then evolve from a representation of the RTT on the previous network, R_{old}, to that of the new network. Depending on the difference ratio between these two values, a variable number of samples will be needed for the sender to have an accurate estimate of R_{new}. The number of samples, denoted n_R^ε, needed to have an estimate within ε of the actual value, that is, $\left|R_r^{n_R^\varepsilon} - R_{\text{new}}\right| \leq \varepsilon$, is

$$n_R^\varepsilon = \left\lceil \frac{\ln \varepsilon - \ln |R_{\text{old}} - R_{\text{new}}|}{\ln q} \right\rceil. \tag{4.14}$$

Assuming that the order of magnitude can vary from the millisecond to the second depending on the network technology and load, it can take up to almost 30 RTTs on the new network for the estimate to be accurate within a 5 % margin.

It is important to note that the TFRC equation (4.1) is directly dependent on this estimate. Given a static $p = p_r$ after the reconnection, as per (4.10), successive samples of the new RTT will refine the estimate which will in turn impact X_{Bps}. As $t_{\text{RTO}} = 4R$, X_{Bps} can be expressed as a function of any of its previous values $T(p, R')$ and the associated RTT estimate R'

$$\begin{aligned} X_{\text{Bps}}^i &= T(p, R_r^i) \\ &= \frac{R'}{R_r^i} T(p, R'). \end{aligned} \tag{4.15}$$

The most useful such relation involves $X_d = T(p_r, R_{\text{old}})$: $X_{\text{Bps}}^i = (R_{\text{old}}/R_r^i)X_d$. Upon reconnection, the new TFRC rate, as compared to that before the disconnection, is thus only dependent on the ratio of the current and previous estimations of the RTT.

Number of Sent Packets When the connection is re-established and the TFRC sender restarts sending packets, it goes through a few phases before being able to resume a rate appropriate to the current network. Depending on the differences between the previous and the current networks' characteristics, the duration (or existence) of these phases will vary. An important factor impacting theses phases is the number of packets that have been sent since the reconnection. In the following, it will be expressed as

$$n^i_{\text{pkts}} = \frac{1}{s}\sum_{j=0}^{i} R^j_r X^j_r, \tag{4.16}$$

where i is the RTT at the end of which the packets are counted and X^i_r is the sending rate during that RTT, as detailed below.

Unused Capacity When connectivity is re-established, two factors can cause the TFRC sender not to fully use the available capacity instantaneously. First, the sender is not directly aware of the reappearance of connectivity. It has to wait for a packet to be acknowledged by the receiver. As the sending rate has been gradually reduced, said packet may not be immediately sent. Secondly, when feedback is received, the sending rate is not resumed directly, but through a phase similar to slow-start (T_c in Figure 4.7).

Additionally, the sender rate will be constrained by the loss event rate p. Due to its history, p will reflect the old network capacity. A better network will therefore not be used at its full capacity until enough loss intervals have been observed. (T'_c in Figure 4.7)

If the sending rate when reconnecting, $X_c = X^{n_D}$, is small, the delay s/X_c between the transmission of two subsequent packets becomes significant. When connectivity is recovered, it can take up to this delay before the first packet is sent. The average idle time after reconnecting can be expressed as

$$t_{\text{idle}} = \frac{s}{2X_c}. \tag{4.17}$$

After this delay, the sender eventually starts increasing the rate. Packets are first sent at rate X_c. Every RTT, a feedback is received with the current value of X_{recv}. According to (4.2), the rate can then be updated to twice this value until X^i_r reaches the rate allowed by the TFRC equation, $X^i_{\text{Bps}} = (R_{\text{old}}/R^i_r)X_d$. During the slow-start RTT i, the sender rate is

$$X^i_r = 2^i X_c. \tag{4.18}$$

Assuming the new network provides the same capacity as the previous one, the average number of packets that could additionally be sent is

$$n_{\text{wasted}} = \frac{1}{s}\left(t_{\text{idle}} \cdot X_d + \sum_{i=0}^{n_{\text{ss}}} R_{\text{new}}\left(X_d - X^i_r\right)\right). \tag{4.19}$$

Parameter n_{ss}, in (4.19), is such that $X^{n_{\text{ss}}}_r \geq X^{n_{\text{ss}}}_{\text{Bps}}$. The development of this inequality using (4.13), (4.15) and (4.18) leads to

$$\frac{R_{\text{new}}}{R_{\text{old}}}2^{n_{ss}} + \left(1 - \frac{R_{\text{new}}}{R_{\text{old}}}\right)(2q)^{n_{ss}} > \frac{X_d}{X_c}, \tag{4.20}$$

which is not linear in n_{ss}. It thus cannot be solved in a purely analytical fashion. In our implementation of the model, we used the Newton-Raphson method (Ypma, 1995) with

$$f(n_{ss}) = \frac{R_{\text{new}}}{R_{\text{old}}} 2^{n_{ss}} + \left(1 - \frac{R_{\text{new}}}{R_{\text{old}}}\right)(2q)^{n_{ss}} - \frac{X_d}{X_c} \quad \text{and} \tag{4.21}$$

$$f'(n_{ss}) = \frac{R_{\text{new}}}{R_{\text{old}}} 2^{n_{ss}} \ln 2 + \left(1 - \frac{R_{\text{new}}}{R_{\text{old}}}\right)(2q)^{n_{ss}} \ln 2q. \tag{4.22}$$

This allowed us to solve (4.20) for n_{ss} in only a few iterations starting from an arbitrary $n_{ss0} = 10$, regardless of the network parameters.

Networks with Larger Capacity The estimate of the loss event rate p is designed to evolve smoothly. This may cause an additional under-usage of the available capacity in case the mobile node connects to a network with a higher capacity $X_{\max}$ than the previous link. Depending on the difference in capacity from said previous network, it may take an unacceptably long time for the sender to eventually occupy the full available capacity. Figure 4.8 shows how TFRC slowly adapts to the new network capacity.

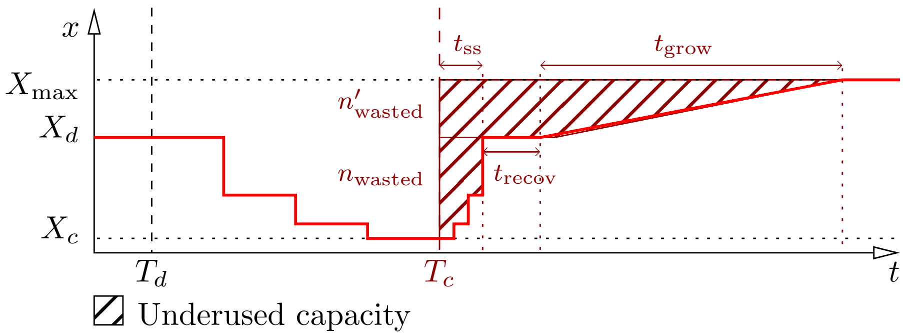


Figure 4.8 After a reconnection, TFRC does not adapt quickly to higher capacities. It slowly uses more capacity as p decreases.

This adaptation time can be decomposed in two periods. First, once the slow-start phase has finished, the sender rate may not immediately start increasing above X_c. Indeed, if there has not been enough packets sent during the slow-start for S_0 to be larger than S_1 in (4.4), p will not decrease. During the loss recovery time, X_{Bps} is kept at value $X_r^i = R_{\text{old}}/R_r^i X_d$. After t_{recov}, when enough packets have been received, p will start decreasing again. During this phase, the sending rate slowly adapts to the available capacity, which is eventually reached after t_{grow}. In addition of n_{wasted}, a further n'_{wasted} packets could be sent. We develop a formulation of this extra capacity wastage below.

The loss recovery time, until the current loss interval contains enough packets not to be ignored in (4.4), is such that

$$\begin{aligned} 0 &< S_0 - S_1 \\ 0 &< w_0 \underbrace{i_0}_{i_{\text{pkts}}^{i\text{recov}}} + (w_1 - w_0)i_1 + \cdots + (w_{i-1} - w_{i-2})i_{i-1} - w_{i-1}i_i. \end{aligned} \tag{4.23}$$

To "compete in the global Internet with TCP," Floyd *et al.* (2008) recommend to take $n = 8$ and the weight vector as $\mathbf{w} = [1, 1, 1, 1, 0.8, 0.6, 0.4, 0.2]$. The formulation of (4.23) can thus be simplified as

$$\begin{aligned} 0 &< n_{\text{pkts}}^{n_{\text{recov}}} - 0.2 \sum_{n=4}^{8} i_n \\ n_{\text{pkts}}^{n_{\text{recov}}} &> 0.2 \sum_{n=4}^{8} i_n. \end{aligned} \tag{4.24}$$

When $n_{\text{pkts}}^{n_{\text{recov}}}$ packets have been sent since the reconnection, p, driven by (4.4), starts decreasing. It is difficult to estimate the i_n as they are dependent on the previous network conditions and specific history. However, assuming a relatively stable network, all i_n would be similar and close to the inverse of p_r, the loss event rate of the previous network. Thus, an estimate of the number of packets that need to be sent before p starts to adapt to the new network conditions can be written as

$$n_{\text{pkts}}^{n_{\text{recov}}} = \frac{1}{p_r}. \tag{4.25}$$

This estimation allows to evaluate the duration of the recovery period, t_{recov}, which exists only if $n_{\text{pkts}}^{n_{\text{recov}}} > n_{\text{pkts}}^{n_{\text{ss}}}$ (that is $t_{\text{recov}} > 0$).

$$\begin{aligned} n_{\text{pkts}}^{n_{\text{recov}}} &= n_{\text{pkts}}^{n_{\text{ss}}} + \frac{Xd}{s} t_{\text{recov}} \\ t_{\text{recov}} &= \frac{s}{Xd}\left(n_{\text{pkts}}^{n_{\text{recov}}} - n_{\text{pkts}}^{n_{\text{ss}}}\right) = \frac{s}{Xd}\left(1/p_r - n_{\text{pkts}}^{n_{\text{ss}}}\right). \end{aligned} \tag{4.26}$$

The additional amount of wasted capacity can be estimated as

$$n'_{\text{wasted}} = \frac{1}{s}(X_{\max} - X_d)\left(t_{\text{idle}} + t_{\text{ss}} + t_{\text{recov}}\right) + \frac{R_{\text{new}}}{s} \sum_{i=0}^{n_{\text{grow}}} \left(X_{\max} - X_r^i\right) \tag{4.27}$$

with

$$X_r^i = \begin{cases} X_d & \text{if } i = 0, \\ \min\left(X_{\text{Bps}}\left(p_{\text{r}} + \Delta p(n_{\text{pkts}}^{i-1}, p_{\text{r}}), R_r^i\right), 2X_r^{i-1}\right) & \text{otherwise.} \end{cases} \tag{4.28}$$

Similarly to (4.19), n_{grow} is the number of RTTs needed to have $X^{n_{\text{grow}}} \geq X_{\max}$.

4.2.4 Model Validation

The model presented in the previous section is verified by comparing numerical results to the output of *ns*-2 simulations for a wide range of network parameters and disconnection durations.

To check the behaviour of TFRC during the disconnected period—(4.5) and (4.7)—as well as the resulting number of lost packets (4.9), 60 s disconnections are introduced after a variable amount of time, for link capacities of 10, 54 and 100 Mbps and for a range of delays (1–100 ms). The number of packets sent after the disconnections is then counted in the simulation trace file. Figure 4.9 shows a comparison of simulation results with predictions from the model for $R = 1$ ms. It confirms that our model exactly predicts the values of the internal parameters of the TFRC sender, and accurately estimates the number of lost packets.

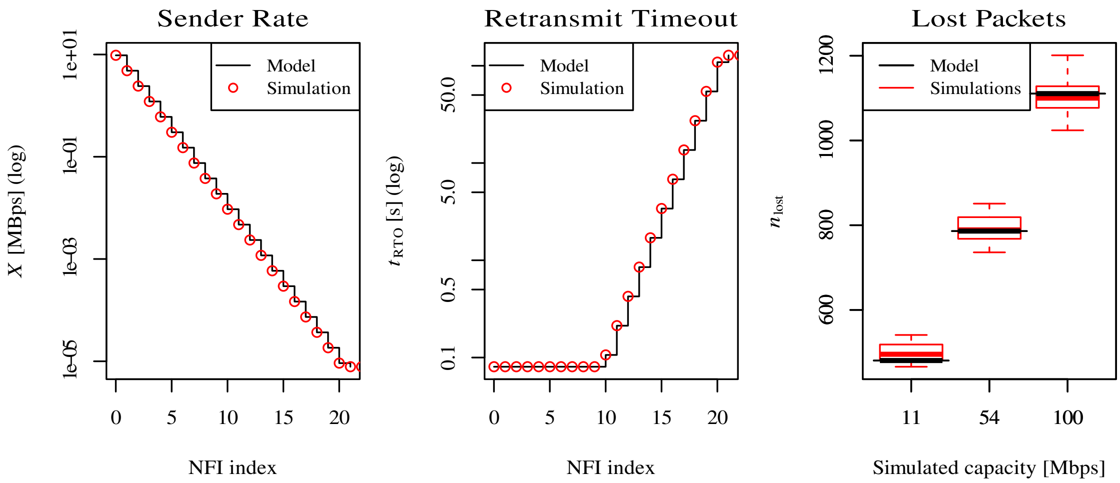


Figure 4.9 Comparison of simulated TFRC's internal parameters, and number of lost packets during disconnections with the model's predictions. Results shown for $R = 1$ ms.

The number of wasted packets, as determined by (4.19) and (4.27) cannot be derived in the same way from the *ns*-2 trace files. Detecting the end of the adaptation periods t_{ss} and t_{grow} relies on comparing the current rate to X_{max}, for which it is impossible to obtain a ground truth from the simulations. It is therefore impossible to identify over which period to count the additional packets which could have potentially been sent, and the results vary depending on the estimate taken for X_{max}. In the following, we take $X_{max} = X_{recv}$ (from Table 4.3); the order of magnitude of the resulting numbers for the wasted capacity are coherent, but the reported value is only indicative, and should not be used in further derivations.

Even though the presented model does not encompass all the details of the behaviour of a real TFRC sender, it has proven to have sufficient prediction accuracy to be used in estimating potential performance gains.

4.2.5 Potential for Improvement

We use our numerical model to determine the performance improvements that can be expected from a better handling of disconnections. The input parameters (X_d and R of both links) for the model are those observed by TFRC when the stationary state has been reached. These are summarised in Table 4.3.

Table 4.3 Network parameters as observed by the *ns*-2 TFRC sender in the stationary phase, reached at T_{stat}. Simulated network characteristics as in Table 2.2 (page 16).

Link type	X_{recv} **[MBps]**	R **[s]**	T_{stat} **[s]**
UMTS	0.044	0.96	660.54
802.16	1.10	0.17	264.14
802.11b	1.27	0.05	50.69
802.11g	4.82	0.04	21.67

Table 4.4 Packet losses and wasted available capacity expected during a MIPv6 handover.

from \ **to**	UMTS	802.16	802.11 b	802.11 g
Packet losses				
UMTS	306	236	226	224
802.16	2760	2614	2614	2614
802.11b	1080	1078	1078	1078
802.11g	2909	2907	2907	2907
Wasted capacity [Number of 500 B packets]				
UMTS	0	8×10^4	3×10^2	1×10^5
802.16	0	5×10^2	2×10^2	1×10^3
802.11b	0	0	1×10^3	5×10^4
802.11g	0	0	0	5×10^3

MIPv6 relies on message exchanges over the network to update the binding with the HA.[7] Therefore, handover delays will vary depending on the characteristics of the current link. The RTT on the new access network is particularly important as it conditions the delay until the MIPv6 binding updates (BUs) are received. Lee *et al.* (2004) thus proposed to use $t_{\text{ho}} = 2.5 + R$ as the time to complete the handover, and during which packets cannot be successfully transmitted to the MN. We use this model as $t_D = t_{\text{ho}}$ in our analysis.

An estimate of the stationary phase value for RTT on the new link is used for R in t_{ho}. It is an over-estimation as the RTT is likely not to be as high upon reconnection as when the full rate is established. However, we use this estimate in our analysis. Therefore, the presented results should be considered an upper bound for the packet loss and lower bound for the wasted capacity.

The number of lost packets and the available capacity wasted during a MIPv6 handover, as predicted by the model, are shown on Table 4.4. These results confirm that the behaviour of DCCP/TFRC can be improved. The next section offers motivations for why this would be desirable and suggests a way to achieve this.

[7] Fast handovers (Koodli and Perkins, 2001) could be used to reduce the disconnection duration. However, packets arriving during the hand-off are buffered at the new access router, which is not desirable for real-time traffic as this would create latency at the application layer upon completion of the handover.

4.3 Freezing the DCCP/TFRC Transmission Upon Disconnections

In this section, we present an enhancement and its implementations to achieve such a betterment. This modification relies on two main additional stages. The sender's state is first frozen just before a hand-off so as not to disrupt its performance, and transmission is suspended. When the handover is complete, the sender is unfrozen. Then, with assistance from the receiver it restores its previous rate and, if possible, probes the new network path for a larger usable capacity.

4.3.1 Rationale of the Improvements

Beyond the packet losses and under-usage of the available capacity, a reduction in the immediate rate is quite detrimental to real-time applications. As previously shown in Section 4.2.2, it can take up to several seconds to restore the rate after the completion of a handover. During this period, applications observe high error rates as they cannot fit the required amount of data units in the rate allowed by the transport protocol, which results in bad quality. In such a situation, restarting from the rate achieved before the handover would enable the application to drastically reduce this period of bad quality.

Considering a communication involving video, the user's experience can be further improved if the new access network has better characteristics, such as a larger capacity. In the new network, the video codec could use a higher encoding rate which would increase the overall quality of experience (QoE). Including a mechanism to probe the new network path can make the new available path capacity available much faster to the application. It could then take advantage of this larger capacity to enhance the overall user experience.

Finally, having information about upcoming handovers, it is also possible to limit, or even nullify, the number of lost packets. Doing so also has the advantage of avoiding the unnecessary use of the old network path to send data which would never be received as the receiver has moved.

When a hand-off is imminent, we thus propose to temporarily suspend the evolution of specific internal parameters. The sender keeps an estimate of the current stationary parameters of the network used to derive the sending rate offered to the application. Further packet transmission is also prevented as the path from the sender to the receiver is known to be temporarily cut. When connectivity is available anew, the rate is restored immediately and adapted as soon as possible to the new network conditions. The congestion control algorithm first allows packets to be sent at the same rate as before. If no error is reported, the sender then tries to probe the network path by increasing its rate. In a way similar to the initial slow-start, the sending rate doubles every RTT until the capacity of the new network has been reached.

4.3.2 New States to Support the Freezing Mechanism

We implement our proposed enhancement to TFRC within DCCP's Congestion Control Identifier (CCID) 3. The operation of the resulting Freeze-DCCP/TFRC is separated into three phases: *Frozen*, *Restoring* and *Probing*. New states are implemented into the sender and receiver to support these. Additionally, new DCCP options are introduced to enable the required signalling for state transition and synchronisation.

Figure 4.10 shows the proposed Freeze-DCCP/TFRC state diagram. The sender has three new states, shown in Figure 4.10a. As most of Freeze-DCCP's operation is driven by the sender, its states are directly named after the three phases. The receiver has two "active" states: *Restoration* and *Probed*. Both *Recovery* states are transient and used to ensure synchronisation with the sender. These are shown in Figure 4.10b.

The following sub-sections detail the signalling options and the evolution of the states, as well as their specific actions throughout the Freeze-DCCP/TFRC phases.

4.3.3 Additional Signalling

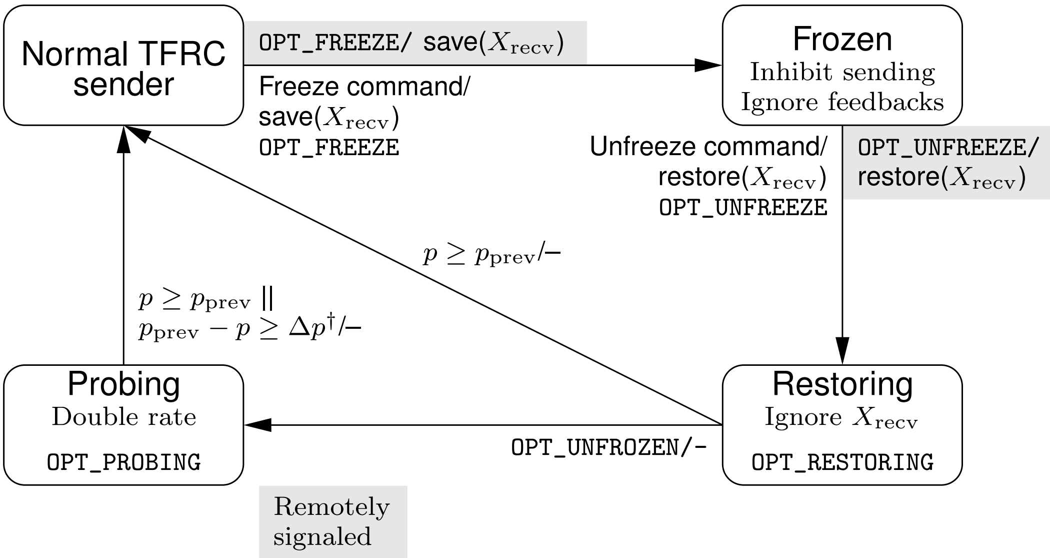
Though window-based flow-control mechanisms have been proposed for TFRC (Lochin *et al.*, 2010), they have not been included in the standard. Thus, unlike Freeze-TCP, it is not possible to freeze a DCCP/TFRC sender by simply reporting a specific value in a feedback message, as for Freeze-TCP (Goff *et al.*, 2000). Additionally, it is desirable to be able to locally suspend the sender. To fully support freezing on both sides, it is necessary to introduce new signalling options, to be carried in the DCCP packet header. Our proposal does not, however, change or extend the format of this header; the options will be gracefully ignored by standard implementations.

In a generic typical case, a mobile node, both sending and receiving over the same DCCP connection, will detect that it is about to lose its current connectivity. In this proposal, packets with an `OPT_FREEZE` option will be sent to the remote peer to suspend its sender operation, then the local sender will be frozen. As the packets carrying the option have to make it to the remote sender, the transport layer has to be informed about upcoming disconnection a short while before it is to happen. A minimum of one RTT is usually considered a reasonable value (Goff *et al.*, 2000). This delay is enough for signalling packets to arrive on time (1/2 RTT later) to prevent the transmission of messages which would have arrived after the disconnection (another 1/2 RTT later).

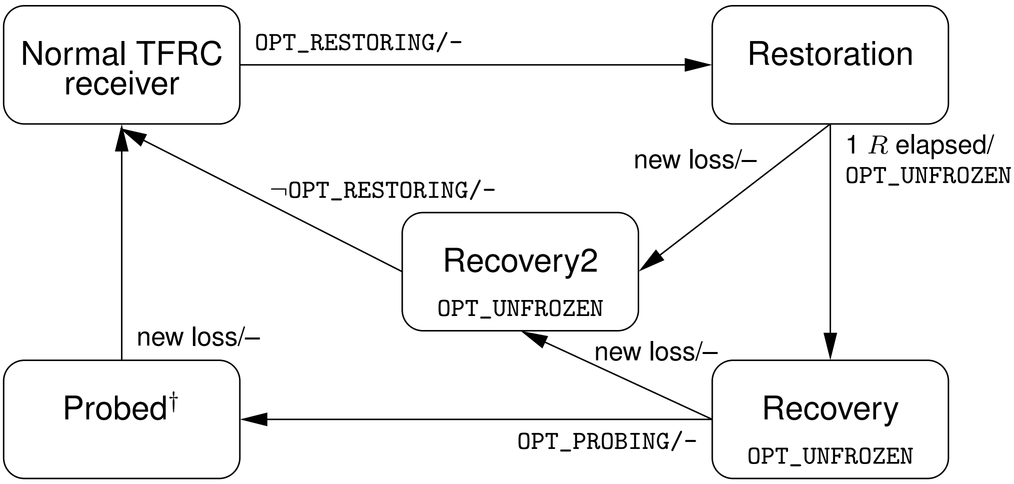
When connectivity becomes available again, the mobile node can restart its traffic, and instruct the remote peer to act similarly by sending packets with an `OPT_UNFREEZE` option.

Additional options are used to support further signalling during the unfreezing phases. The sender uses `OPT_PROBING` and `OPT_RESTORING` to indicate the state that it is currently in, while the receiver sends an `OPT_UNFROZEN` to signal that it is ready for the Probing phase.

As DCCP is an unreliable protocol, option-carrying packets can be silently lost. Extra care must be taken to ensure both peers are synchronised. This can be done by exchanging options in a redundant manner. The naive approach of adding those to every outgoing packet is chosen here. Depending on the application, this however risks consuming too much capacity and reduction of the option frequency could be considered.



(a) Freeze-DCCP/TFRC sender



†When a packet is lost, the receiver computes and reports a p equivalent to the currently observed X_{recv}.

(b) Freeze-DCCP/TFRC receiver

Figure 4.10 Additional states and option exchanges to support Freeze-DCCP/TFRC (transitions are labelled as Condition/Action). The sender (a) can be instructed to freeze or unfreeze either locally or by the remote peer. The receiver (b) does not have to enter a Frozen state, but must perform some specific tasks during the Restoration and Probed phases. Options can signal both/either the remote sender and/or receiver.

4.3.4 Frozen Phase

When instructed to freeze, either locally or by the remote peer, the sender enters the Frozen state. In this state, all data transmission ceases. This ensures that no packet will be lost. It in turn guarantees that the loss event rate calculated by the receiver will be kept unmodified. The receiver does not need any specific state to support this phase.

The disconnection may however not happen right after freezing, and additional feedback from the receiver may arrive at the sender. Parameters such as the RTT R, or the receiver rate X_{recv} risk being updated. Thus, while in the Frozen state, the sender ignores all feedback messages. When entering this state, it also saves the value of X_{recv} as it will be locally modified on every expiration of the `nofeedback` timer.

To efficiently address longer disconnection periods which may occur, for example, in Delay-Tolerant Networks (DTNs), it is advisable to additionally increase the connection timeout. Indeed, disconnections longer than 8 minutes may result in the frozen socket being prematurely closed.[8]

4.3.5 Restoring Phase

After receiving a local unfreeze instruction or the `OPT_UNFREEZE` option, the sender enters the Restoring state. It first restores X_{recv}. The send timer is then reset to resume packet transmission. As the parameters are the same as before the disconnection, the sending rate will be restored to its previous value.

At the same time, it is no longer necessary to completely ignore feedback from the receiver. It is however needed to keep ignoring the X_{recv} reports. Indeed, the receiver rate is measured over at least one RTT. The first feedback packets are likely to cover part of the disconnected period resulting in an incorrectly low value for X_{recv}. Using such value may create instabilities in the sending rate as it is bound by $2X_{\text{recv}}$ as per (4.1).

When in the Restoring state, the sender adds an `OPT_RESTORING` option to all its outgoing packets to put the receiver into the Restoration state. The Restoring phase ends when the loss event rate increases or an `OPT_UNFROZEN` option is received. This option is added by the receiver after a complete RTT has elapsed, thus signalling that it is no longer necessary to ignore the value of X_{recv} as it will now correctly reflect the receiver rate.

4.3.6 Probing Phase

Standard TFRC quickly reacts to a reduction in the available capacity by responding promptly to an increase in the loss event rate. The `conservative_` mode outlined by Bansal *et al.* (2001) further increases this response. Conversely, after idles periods, Kohler *et al.* (2008) propose to increase the sending rate back to the previously supported max-

[8]According to Kohler *et al.* (2006b), a socket in the *Respond* state waits a maximum of four Maximum (TCP) Segment Life for packets before resetting the connection. The Linux 2.6 implementation generalises this idle timeout to the entire lifespan of the socket.

imum at an increased pace by quadrupling the rate every RTT.[9] There is however no mechanism to quickly adapt to *better* network conditions. In the Probing state, our sender checks for such improvement in the new network. This is done only if no loss has occurred during the Restoring phase. The sender uses the `OPT_PROBING` option to inform the receiver of its new state. Upon receiving this option, the receiver enters the Probed state.

This phase is similar to a slow-start. Every RTT, the sending rate is doubled. When a loss is detected while it is in the Probed state, the receiver reinitialises its loss history to match the last measured rate. It first computes a packet loss rate p equivalent to the observed receiver rate X_{recv}. It then reinitialises a complete history of n loss intervals of the calculated size.

As p is completely recomputed by the receiver on the first loss, it can be larger, lower or even equal to its previous value. The exit criterion for the probing phase is therefore based on the expected evolution of the reported loss event rate, as expressed by (4.10). In a loss-less period, p will never increase. With a growing loss interval, it will however keep decreasing slightly. The sender should thus exit the Probing state if

$$\Delta p \notin \left]\Delta p_{\min}(X_{\text{Bps}} \cdot R, p_{\text{prev}}); 0\right[\,, \tag{4.29}$$

following (4.11). The absence of the `OPT_PROBING` option on new packets will in turn take the receiver out of the Probed state.

It may happen that the sender-recomputed p lies in the acceptable range of variation. In this case, the sender cannot detect that the Probing phase should be ended. Some more losses will however be generated during the next RTT. These losses will prevent p from changing during the next report, thus properly ending the Probing phase as per the previous criterion.

4.4 Performance Evaluation

This section presents an evaluation of the enhancements proposed in the previous section. It first compares, in *ns*-2 simulations, the behaviour of Freeze-DCCP/TFRC with that of the unmodified version. It then shows that the proposed mechanism still retains a satisfying level of fairness to TCP flows. Finally, it shows in a real experiment, based on a Linux implementation, that our proposal is well suited to improve the QoE of a live video stream experiencing multiple handovers between heterogeneous technologies.

4.4.1 Realistic Handover Scenarios

Simulations were run with *ns*-2.33. Additional modifications have been made to the TFRC sender of Mattsson (2004)'s DCCP module to implement the clarified loss average calculation of Floyd *et al.* (2008). The `DCCP/TFRC/Freeze` agent has been implemented by deriving the DCCP/TFRC C++ class to add the freezing mechanisms described above.

[9] This draft proposal considers application-limited rates rather than disconnections; it also does not restore the rate at once as our Restoring phase does.

As only the impact of disconnections on the transport layer is relevant to our study, it is irrelevant to simulate all the underlying wireless networks as such. Rather, simple wired topologies are used, as suggested by Gurtov and Floyd (2004). All wireless links were modelled as duplex links, even the 802.1x ones. This may not be correct for bi-directional data scenarios. Such scenarios, however, are not considered here.

A router is placed between the sender and the receiver. Such a topology allows to transparently disconnect the link between the router and the receiver without preventing the sender from trying to transmit packets during the disconnections. `DropTail` queues with the default buffer size (50 packets) are used.

Disconnections are simulated by manipulating the routing model of *ns*-2 with the `$ns_ rtmodel-at` function. The time of the hand-offs is chosen, once the system is in a stationary state, from a uniformly distributed variable over a time period of four RTTs. The generic behaviour of both standard TFRC and our variant is thus captured. Link characteristics are modified using the `$ns_ bandwidth` and `$ns_ delay` commands. The simulations were ended after the rates have settled on the new network. The presented results have been averaged over 20 runs.

The Freeze-enabled DCCP agent is instructed to suspend its connection locally (*i.e.*, not using the `OPT_FREEZE` and `OPT_UNFREEZE` options). The `freeze` command is given one RTT before the disconnection is scheduled to happen (as suggested by Goff *et al.*, 2000). The `unfreeze` instruction is given 0.1 ms after the network link is reconnected.

The number of losses upon reconnection, as well as the wasted capacity, is shown on Table 4.5. As Freeze-DCCP/TFRC did not lose any packet, this information is omitted. The wasted capacity has been estimated by comparing TFRC's actual rate X to what is achievable in the steady state (Table 4.3), then converted in number of 500 B packets. Figure 4.11 illustrates these figures by comparing how both regular DCCP and the Freeze-enabled version perform in key example scenarios.

Two cases stand out in Table 4.5 where the average amount of wasted capacity is larger with our proposal than with standard TFRC. These correspond to pathological cases for our *ns*-2 implementation where either the recovery or probing phase experience losses too early. The rate reported by the receiver does not cover a full RTT (802.11g to 802.16), or the probing phase terminates before having discovered the network capacity (UMTS to 802.11b). Future work should address this type of situation by using other or additional metrics, rather than just losses, to drive the Restoration/Probing phases.

Overall, Freeze-DCCP/TFRC quickly restores an acceptable rate for the application, and greatly reduces the overall under-usage of the available capacity upon reconnection. It also prevents traffic after the hand-off to unnecessarily use the rest of the network.

4.4.2 Fairness to TCP Flows

TFRC was designed to quickly respond to reductions of the capacity. The restoring and probing features of Freeze-DCCP/TFRC, however, aggressively use and test the network. Though Briscoe (2007) argues that usage fairness should not be based on rate comparisons, such an approach is still commonly accepted. TCP-friendliness is also a key property of

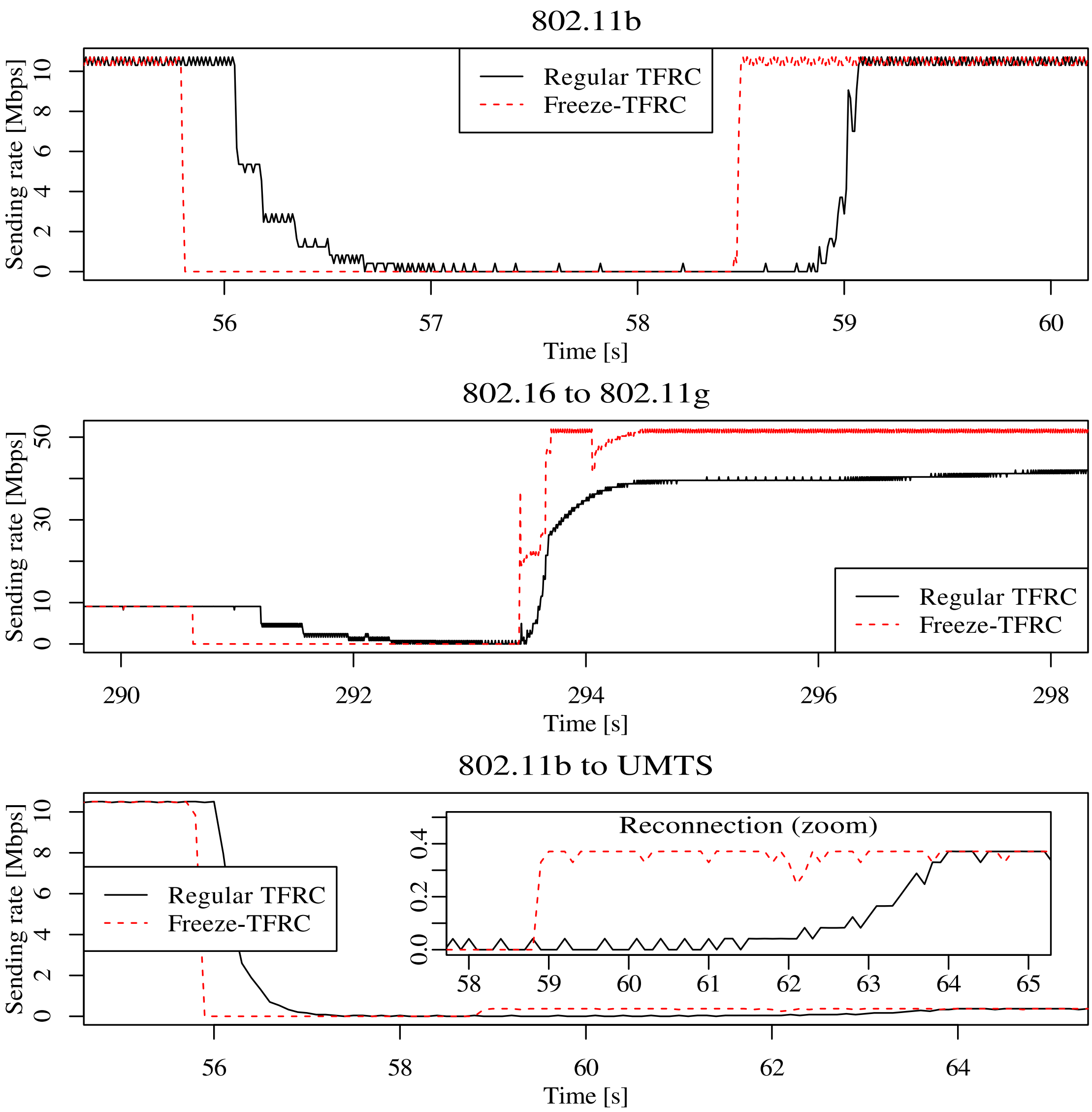


Figure 4.11 Comparison of the rate of DCCP/TFRC and the Freeze-enabled version in typical examples of MIPv6 horizontal or vertical handovers.

Table 4.5 Simulated MIPv6 handovers performance impact for DCCP/TFRC (top cell) and Freeze-DCCP/TFRC (bottom cell).

from \ to	UMTS	802.16	802.11 b	802.11 g
Packet losses (DCCP/TFRC only)				
UMTS	253.3	269.8	273.6	275.4
802.16	1732.3	1734.6	1734.6	1734.6
802.11b	856	855.5	855.3	855.3
802.11g	2470.9	2470.4	2470.2	2470.1
Wasted capacity [Number of 500 B packets]				
UMTS	50.5	54018.05	2190.45	92156.1
	13.4	3607.9	9342.75	89328.6
802.16	12.45	1827.95	603.05	4185.75
	5	591.15	150.9	1520.35
802.11b	150.45	28314	2101.75	57970.65
	0	15278	47.45	1045.05
802.11g	42.5	2104.3	943.4	4313
	0	7172.75	46.5	188.45

TFRC. It is therefore important to check that our additions do not make the protocol too greedy.

The criterion to evaluate TCP-fairness $f(t)$ is the ratio of the average capacity occupation of Freeze-DCCP/TFRC $C_D(t)$ to that of concurrent TCP flows $C_T(t)$,

$$f(t) = \frac{C_D(t)}{C_T(t)}. \tag{4.30}$$

The samples, taken after the disconnection, have been averaged over 100 s, discarding the initial rate settlement period.

Table 4.6 shows the average fairness of a Freeze-DCCP/TFRC flow to a concurrent TCP stream, as observed after the reconnection for the studied handover scenarios. The proposed improvement appears to be reasonably fair to TCP flows in various scenarios including vertical handovers to technologies with higher or lower capacities. In some cases it is even too fair, not competing aggressively enough for the network. A similar behaviour has however also been observed for the regular DCCP/TFRC.

4.4.3 QoE of Mobile Video Streaming

To explore the actual performance improvements of our proposal, we used an OMF testbed (Rakotoarivelo *et al.*, 2010) to emulate vertical handovers and observe the impact

Table 4.6 Fairness comparison of the Freeze-enabled proposal to TCP after a handover. Values in the range [0.5; 2] are considered "reasonably fair" (Floyd *et al.*, 2008).

from \ to	UMTS	802.16	802.11 b	802.11 g
UMTS	0.6	0.3	0.2	0.1
802.16	1.6	1.3	1.1	0.9
802.11b	1.3	1	0.9	0.7
802.11g	1.5	1.2	1	1.1

on the quality of a video stream. An implementation of Freeze-DCCP/TFRC has been developed in the Linux kernel,[10] and used for this experiment.

The scenario is depicted in Figure 4.12. We consider a common scenario where a user, initially at home (t_0), receives a video stream on their mobile terminal connected to their home Wi-Fi network (1 Mbps). They decide to get a coffee from the corner shop. On the way there, the mobile terminal loses its connectivity to the home network, and hands off to the 3G network (t_1; 500 kbps). The coffee shop has a public wireless network (700 kbps), to which the device connects when it gets in range (t_2). With their coffee in hand, the user then heads back home, losing connectivity to the public Wi-Fi network and performing a new handover to 3G at t_3 before finally reconnecting to their home network at t_4. Throughout the streaming period, the device thus goes through several handovers between various wireless networks with different capacities and delays.

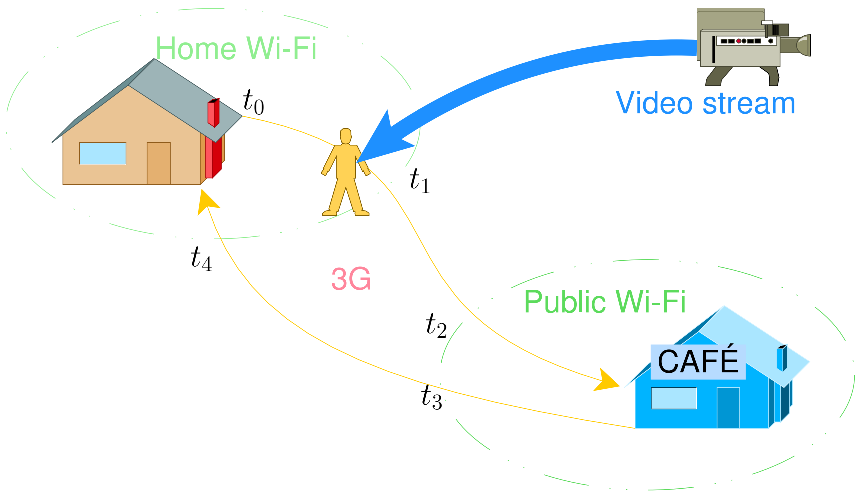


Figure 4.12 Scenario for the evaluation of Freeze-DCCP/TFRC improvements on application quality. A user viewing a video stream on their mobile device goes out for a coffee then comes back home. In the process, different networks with various capacities are visited.

[10] We used the Net:DCCP tree (http://eden-feed.erg.abdn.ac.uk/cgi-bin/gitweb.cgi?p=dccp_exp.git;a=summary) as a starting point. At the time of this work, it was forked off of vanilla version 2.6.34-rc5. A Git branch containing these modifications is available at http://github.com/shtrom/linux-2.6/tree/freezedccp.

In this experiment, a video file, encoded and packetised using the H.264 codec with a 1 Mbps bit-rate, is sent using a specially patched version of Iperf[11] to a custom receiver application. Both endpoints provide information about the sending or receiving rates. The custom receiver identifies lost packets and computes a moving average of the peak signal-to-noise ratio (PSNR) over 24 frames ($\simeq 1$ s). Both sender and receiver have been instrumented with OMF Measurement Library toolkit (OML; White *et al.*, 2010). This allows them to report readings of these metrics in real time for analysis or display.

Our OMF testbed currently only supports 802.11-based wireless networks. However, only a limited set of parameters of the underlying network is relevant at the transport layer. Therefore, as for the previous section, we follow the suggestion of Gurtov and Floyd (2004) and emulate the conditions of wireless technologies in different networks by shaping the available capacity (using Linux' traffic control tools, Hubert *et al.*, 2004) and regulating forwarding delays (using NetEm; Hemminger, 2005).

The results, averaged over 8 runs of our scenario, comparing the PSNR of the video stream when using Freeze-DCCP/TFRC to that with the standard version, are shown in Figure 4.13. They show that the use of Freeze-DCCP/TFRC, as it is able to adapt much faster and use close to the full available path capacity to carry application data, results in a reduced but stable QoE when on those networks which can't support 1 Mbps streams. In comparison, the PSNR of a video stream supported by the regular DCCP/TFRC reduces to a minimum (a PSNR of 7 dB is that of a purely random image), and takes a long time (up to the complete visit duration of a network) to restore to a better level. In addition, Freeze-DCCP/TFRC does not suffer from the oscillations which appear for the standard version when visiting the Café's Wi-Fi network. We hypothesise that the probing mechanism finds the capacity of the new network more accurately, and its estimates are not biased by measurements from the previous network.

4.5 Conclusion and Future Work

In this chapter, we have first identified the issues that TFRC faces in mobility situations. We have numerically modelled the losses and subsequent under-usage of the available capacity that it experiences in those cases. This model allowed us to evaluate the performance improvements that could be expected from a system with a better awareness and handling of disconnections. We thus proposed Freeze-DCCP/TFRC, an extension of the TFRC congestion control mechanism used by DCCP, to approach these possible performance gains. This proposal is aimed at uses of DCCP in situations where network connectivity may periodically not be available for varying periods of time, and the access networks' characteristics may widely vary between disconnections.

Freeze-DCCP/TFRC was both implemented in *ns*-2 and Linux. Simulation results have shown that it is possible to prevent handover-induced losses, to restore the rate faster when reconnecting to a link with lower or similar capacities and to adapt more quickly to higher capacities. Additionally, we confirmed that our proposal maintains the important TFRC's

[11] This version, supporting both DCCP and OML, is available from `http://www.nicta.com.au/people/mehanio/freezedccp#iperf-dccp-oml`.

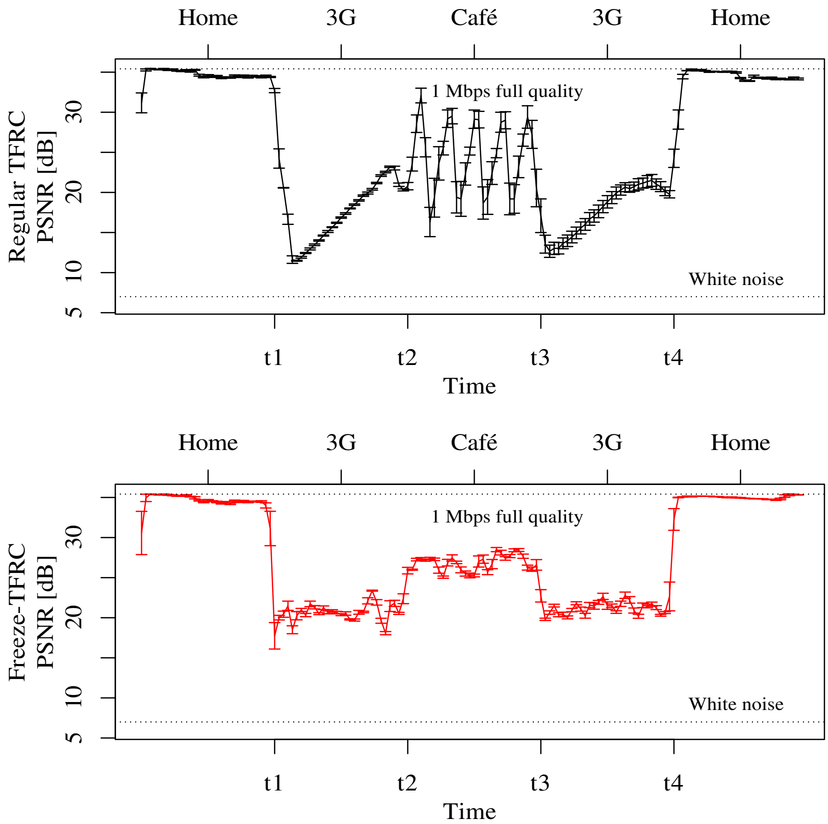


Figure 4.13 PSNR comparison for a video stream using TFRC or Freeze-TFRC in mobility situations. Averages over 8 runs; error bars indicate the standard error.

property of being fair to concurrent TCP flows. We have also experimentally shown that our proposal can significantly improve the performance quality of streaming applications when disconnections are predictable, for example, for IP mobility or, more generally, DTN. Though the proposed modifications were designed with real-time applications over DCCP in mind, they are versatile enough in terms of rate adaptation and packet loss avoidance to also benefit other types of traffic.

Additional work is however needed for the proposed extension to be used in real deployments. First, more attention should be paid to the robustness of the state machine. Particularly, scenarios with both a mobile sender and a mobile receiver may create situations where packets carrying the synchronisation options get lost, thus putting the communication socket in an impossible, and currently irremediable, state. Second, it would be desirable to decouple the freezing mechanism, which caters for hand-off-induced disconnections, and the probing phase, which deals with heterogeneous paths. Indeed, only the latter is needed for make-before-break handovers. Third, the current probing mechanism assumes packets are lost only when the capacity of the new network is reached. Other methods of detecting that the current rate matches the available path capacity should be explored. Both in-band solutions, such as Lin *et al.* (2006)'s MBTFRC, and out-of-band ones (*e.g.*, based on 802.21 and provided by our cross-layer framework) could be considered.

Chapter 5

Accuracy of a Measurement Instrumentation Library

5.1 Introduction

Our proposed cross-layer framework relies on being able to obtain current performance indicators from the local stack, as well as metrics about the reachable access networks. Such data can be used immediately, or stored alongside other miscellaneous contextual information to support prediction algorithms (*e.g.*, Rathnayake and Ott, 2008; Petander, 2009) and more informed decisions. Figure 5.1 illustrates the data collection path of our framework, which is the topic of this chapter.

As mentioned in Section 2.4.3 (page 43), tools such as Iperf or `tcpdump` can be used to probe or observe some network characteristics in order to measure the desired metrics. However, this type of stand-alone tools usually reports their measurements in incompatible application-specific formats. This limits their usability for our framework as it entails a need for on-line data preprocessing mechanisms. In addition, some performance indicators are directly computed by the elements of the stack for which they are relevant. Therefore we argue it is more appropriate to directly obtain them from the elements which already measure them as part of their normal operation (*e.g.*, the transport protocol's round-trip time, RTT), rather than trying to recompute or estimate metrics. The Web100 instrumentation (Mathis *et al.*, 2003) provides an application programming interface (API) exposing internal parameters of the transport protocols. However, such a convenient API is rarely available natively at other layers.

The OMF Measurement Library (OML; White *et al.*, 2010) is a lightweight instrumentation and measurement collection system for distributed network experiments. It provides a simple API allowing a developer, or programmer in charge of instrumenting a third-party tool, to report any type of information using the OML protocol. All data reported this way is stored in a timestamped database, thus simplifying unified data access, aggregation

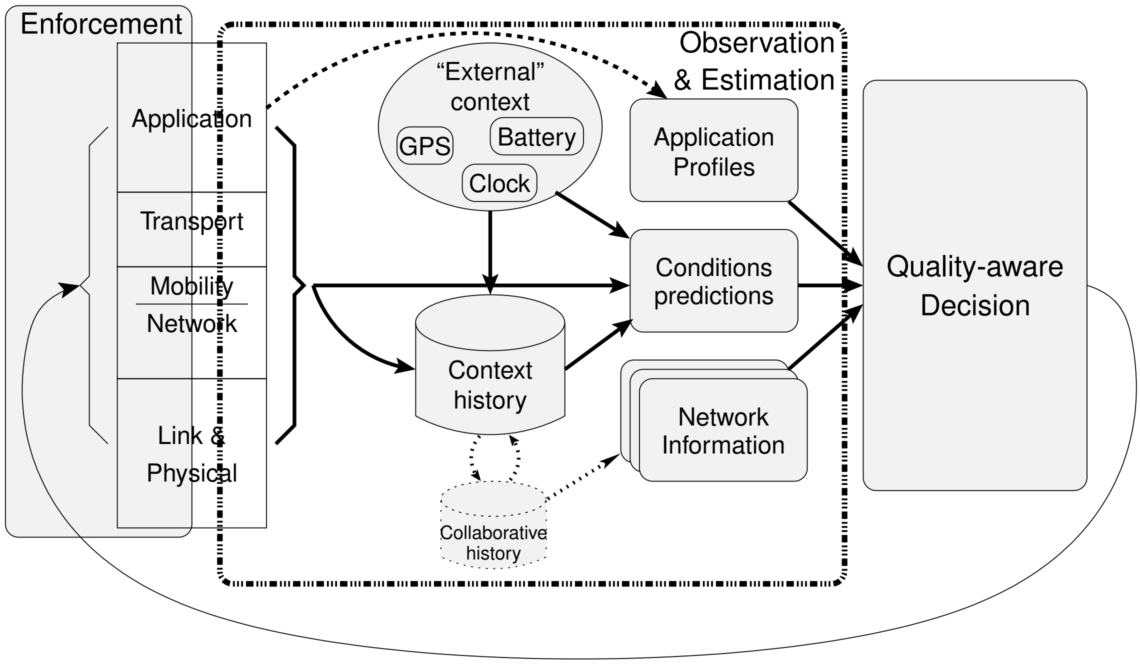


Figure 5.1 Relation of the contribution presented in this chapter to the cross-layer framework of this thesis. Bold lines and non greyed-out components are the current focus.

and usage. We therefore propose to use this library as the information-exchange bus of our cross-layer framework, collecting performance indicators and contextual information from the various elements of the system, and making it available to the decision mechanism in a consolidated format.

However, as the OML reporting is done by the application itself, this may disrupt its performance, or even proper functionality of the device by using too many system resources. It is therefore important to characterise the influence of this software reporting tool on the observed performance, in order to assess the accuracy of the measurements it collects. In this chapter, we develop an experimental and analytical process to identify potential differences in behaviour between different flavours of selected network measurement tools. This allows us to conduct a comprehensive study of OML in terms of the potential biases it introduces in measurements it collects, and its impact on the performance of tools that use it for instrumentation.

OML is described in more details in Section 5.2.[1] We choose to evaluate its impact on stand-alone popular tools Iperf (version 2.0.5) and libtrace (Alcock *et al.*, 2010) (version 3). This instrumentation is described in Section 5.3.1. We design a series of experiments (Section 5.3.2) to determine the effect of OML instrumentation. We then investigate the impact of several experimental factors on various dependent variables for both application performance (Iperf) and accuracy of measurement (libtrace), using analysis of variance (ANOVA) techniques, for which the results are presented in Section 5.4. In Section 5.5, we discuss the operating ranges and scenarios where the effects of OML are significant and

[1]We worked with OML version 2.5.0, the most current at the time of this study.

where they are negligible. Our results support the use of OML as the collection system for our framework. Moreover, we find that even a naive instrumentation using OML can perform equivalently to a sophisticated hand-coded measurement collection strategy; to better cope with those cases where the measurement rate is too high, we also suggest recommendations on how to properly instrument applications and set up the collection path for our framework in Section 5.5.4.

5.2 Presentation of OML

OML is a multithreaded instrumentation and measurement library, which was first developed as a component of the cOntrol and Monitoring Framework (OMF; Rakotoarivelo *et al.*, 2010), but is now a stand-alone open source software[2] which can collect any type of measurement from any type of application and store them in a unified format. Measurement reporting via OML can be added alongside original reporting mechanisms or as their replacement. A unified approach using OML to collect measurements allows effortless correlation of data from different distributed sources to investigate network anomalies, or test research hypotheses or developed prototypes.

OML has three components that allow a user to automatically generate and collect measurements. First, a developer defines *measurement points* (MP) within their applications or services. An MP is an abstraction for tuples of related metrics which are reported ("injected") by the application at the same instant. At run-time all or a subset of these MPs can be requested to generate *measurement streams* (MS). Samples from unselected MPs are discarded, as they are deemed irrelevant for the current experiment. MSs can be instructed to report samples from their MPs as soon as a specified number (one or more) of samples has been injected, or compute an aggregate at a defined frequency. Before being streamed towards repositories to be stored for later analysis, MSs can also be further processed.

This processing is done through OML's filtering mechanism, which extends the periodic aggregation mentionned above. Additional functions can be applied on some of the fields of an MS to format the data or compute more specific metrics. For example, for an application reporting the size of each packet it receives in an MP, a filter may be used to sum these samples over a 1 second period to provide an estimate of the immediate goodput. This goodput can then be further processed by an averaging filter with a 1 minute period. OML provides some generic filters such as the aforementioned, but also exposes an API so more specific filters can be developed.

Figure 5.2 shows an example OML data path. An application injects measurement into three MPs. At run-time, the tuples generated by injections in the MPs are combined in order to form five MSs. These newly created streams are then filtered, and the results are directed to either of two different collection servers, or a local file. The right part of Figure 5.2 represents the server side where the OML server serves as a front-end to a database.

[2] http://oml.mytestbed.net

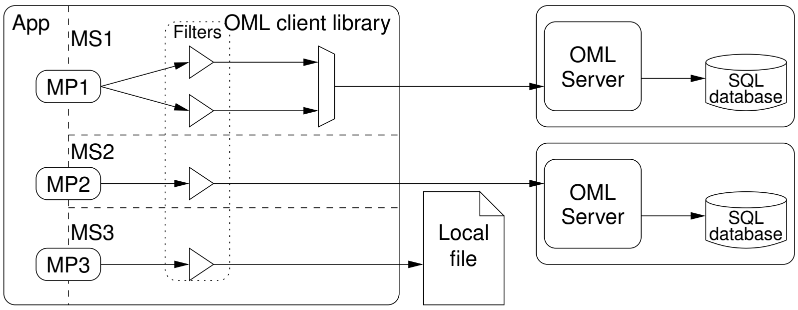


Figure 5.2 Measurement data path in OML: Three measurement streams (MS) are filtered to generate measurement streams (MS) which get stored at different locations.

By default, MSs are sent reliably to the server using TCP. However, if the network path used for reporting experiences transient losses, or cannot provide a sufficient capacity, the buffers of the OML client may fill up, and samples could be lost before having been sent to the server.

OML also provides a timestamping mechanism based on each reporting node's time (`oml_ts_client`). Each server remaps the MSs they receive to an experiment-wide timebase (`oml_ts_server`) which allows some time comparisons to be made between measurements from different machines. This mechanism however does not remove the need for a good time synchronisation between the involved experimental nodes.

OML has been integrated in many applications, such as traffic generators, passive network measurements, GPS coordinate recorders, and pressure/temperature sensor monitors.[3]

5.3 Method

In order to evaluate the effects of OML 2.5.0, we have designed several experiments. We compare performance indicators and assess measurement accuracy between the original and OML-instrumented software tools. This section first presents our modifications to the Iperf traffic generator and the libtrace packet capture library. We then describe our experiment designs and procedures.

5.3.1 Instrumented Tools

For this study, we instrumented tools for network measurement (network probing, packet capture). We use Iperf as the traffic generator and the canonical `tcpdump`, as well as a more recent packet capture library, libtrace (Alcock *et al.*, 2010), for network measurements. In addition, we also instrumented a system metrics-measurement library in order to allow us to observe the potential additional load induced by the use of the tools we are characterising. This section details these instrumentations.

[3] http://oml.mytestbed.net/wiki/omlapp

Network Probing: Iperf

Iperf allows to test the characteristics of a network path using either TCP or UDP. Its code is multithreaded to limit the impact of reporting—either on the console or to a CSV file—on the high-speed generation of probe packets. Iperf can report a number of metrics depending on the transport protocol in use. For TCP, only the transferred size, from which the throughput is derived, can be observed. For UDP, packet loss and jitter information can also be reported. The periodicity of Iperf's reports is configurable from once for an entire run to as frequently as every half a second. The internal aggregation function depends on the metric: the transferred size and losses are summed, while the latest value to date is reported for the jitter.

We have instrumented version 2.0.5 of Iperf to support reporting via OML.[4] Two separate modes of operation have been implemented in the form of new reporting styles, *legacy* (`iperf -y o`) and *advanced* (`iperf -y O`), which differ in the amount of processing that is done in the application. Table 5.1 summarises the performance metrics directly reported by the different flavours of Iperf used in this study. Figure 5.3 shows the main traffic-generating loop of Iperf, and how the OML instrumentation has been integrated into it.

Table 5.1 Summary of the information reported by the various flavours of Iperf considered in this study.

flavour \ **transport**	Reliable stream	Unreliable datagrams
Vanilla	Transferred size, Throughput,	*idem* + Losses, Jitter
OML legacy	Transferred size	*idem* + Losses, Jitter
OML advanced	Packet ID and size, emission and reception timestamps	

In the legacy mode, the aggregation of the measurements is done using Iperf's native code, and the periodic reports are sent out through OML via three MPs: `transfer` for the size, `losses` for lost and sent datagrams, and `jitter` for Iperf's implementation of (2.1). In the advanced mode, Iperf directly reports information about each packet sent or received via OML, in the `packets` MP which contain identification, size and both sent and received (if relevant) times for each packet, down to the microsecond. The advanced mode is more in line with OML's approach, where the measured data is reported verbatim by the application and all processing and consolidation is done through filters, thus allowing more specific treatment. Noting that, in most of the literature based on Iperf, there is a lack of precise reporting of versions, platforms and parameters, we also implemented MPs to report such ancillary information about the experiment as the version numbers and command line arguments.

[4]The latest instrumented code is available at http://omf.mytestbed.net/projects/Iperf/repository/show?rev=oml/master

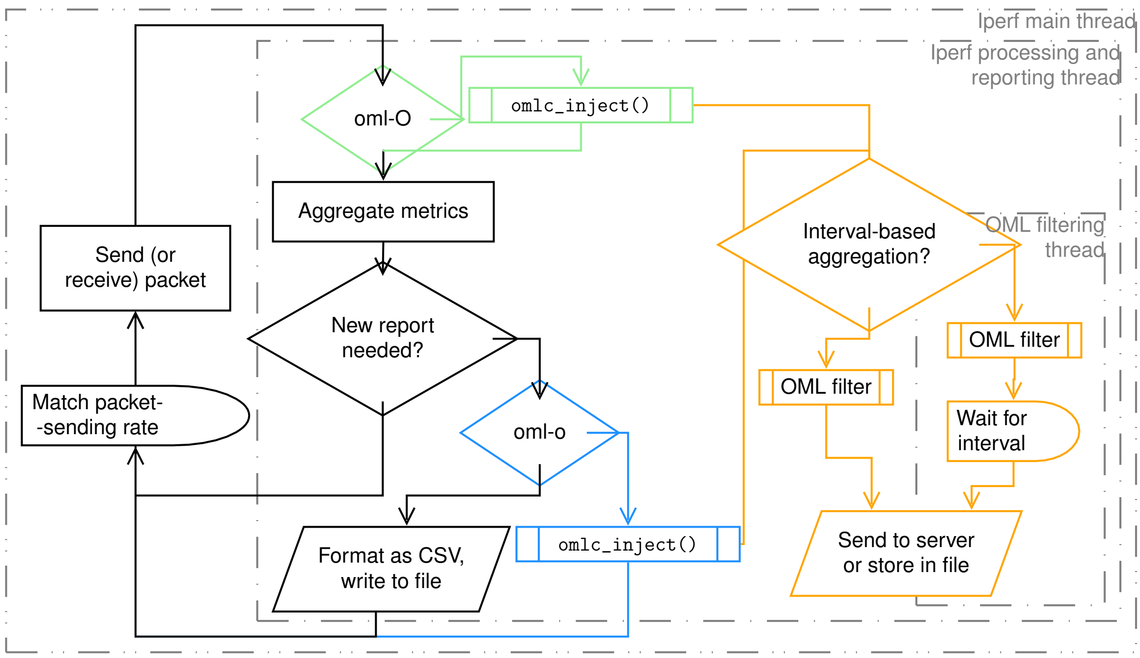


Figure 5.3 Iperf main loop (black) and OML instrumentation additions (colours). OML reporting is implemented in two sections. The legacy mode (oml-o, blue) reports aggregate metrics computed by Iperf, while the advanced mode (oml-O, green) reports each packet. In both cases, the data can be filtered before being sent for storage (orange). Iperf can delegate reporting to a second thread if activated, while OML uses a filtering thread for time-based aggregates (*e.g.*, per-second averages).

The code we modified to integrate OML is part of the main sending/reporting loop of Iperf. It therefore has a potential to impact the application's performance. Section 5.3.2 describes the experiment we designed to evaluate this effect.

Packet Capture: `tcpdump` and libtrace

In this study we focus on the use of both the libpcap in its most simple way with the `tcpdump` tool and our implementation of a packet-capturing application with OML reporting written as a wrapper around libtrace functions, `oml2_trace`. In our case, both libraries use the Linux Socket Filter (Insolvibile, 2001) to get packets from the kernel.

The main difference between these tools concerns the range of captured frame information. In the case of `tcpdump`, the binary Ethernet frame is dumped in its entirety (up to the maximum `snarflen`, as specified by the experimenter). With `oml2_trace`, the information is read from the protocol headers and injected into different MPs, thereby giving the experimenter more control on what is collected. Both applications also support Radiotap pseudo headers to provide information about wireless channels.[5]

Though OML provides timestamping on its own, packets captured from the Linux Packet Filter carry precise timing information. As version 2.5.0 of OML does not provide a mech-

[5] http://www.radiotap.org

anism for the application to set the `oml_ts_client` of the reports, the capture timestamps are included as fields of the measurement points.

Table 5.2 summarises how the considered tools make these metrics available to the rest of the system through OML. It highlights the much finer granularity available from `oml2_trace`.

Table 5.2 Comparison the way information is reported by both packet-capturing applications.

<table>
<tr><th>Protocol</th><th>Fields</th><th>tcpdump</th><th>oml2_trace</th></tr>
<tr><td>Ethernet</td><td>Packet ID, MAC addresses, etc.</td><td rowspan="5">Binary dump (possibly truncated to snarflen)</td><td rowspan="2">radiotap MP</td></tr>
<tr><td>(Radiotap)</td><td>Wireless channel characteristics</td></tr>
<tr><td>IP</td><td>ID, length, addresses, etc.</td><td>ip MP</td></tr>
<tr><td>TCP/UDP</td><td>ID, ports, length, etc.</td><td>tcp or udp MPs</td></tr>
<tr><td>Timestamp</td><td></td><td>All MPs</td></tr>
</table>

Resource Usage: Sigar

Sigar (System Information Gatherer And Reporter) is a library which provides cross-platform system performance metrics.[6] An OML instrumented tool, `oml2_nmetrics`, has been developed based on this library in order to enable basic system monitoring.

This tool can report system information such as CPU load, memory used and network operation as well as per-process state and system usage. In our study, we only use this tool to observe the changes in system load following the introduction of OML instrumented applications.

5.3.2 Experiments

We use the simple topology illustrated in Figure 5.4 for numerous experiment trials with varying applications and parameters on the nodes. We used the NetEm emulation software (Hemminger, 2005) to vary the available link capacity between the sender (*Snd*) and the receiver (*Rcv*). We run two main types of experiments to characterise the OML instrumented tools and their vanilla equivalents.

To provide a reference for meaningful comparisons, we always run a `tcpdump` instance on both *Snd* and *Rcv* to let us compute, *a posteriori*, the actual rate and jitter which we assume to be accurate estimates of the ground truth. We can therefore observe the

[6] http://support.hyperic.com/display/SIGAR/Home

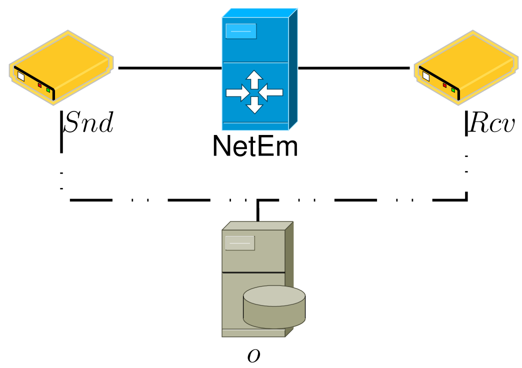


Figure 5.4 Experiment topology. An Iperf sender *Snd* generates traffic towards a receiver *Rcv* on a link capacity-shaped using NetEm. When OML applications are used, measurements are sent over the control network to OML server o.

trials both through the OML instrumented tools and the `tcpdump` packet traces, using the following main metrics:

rate R_X^S observed by tool X (i for Iperf, t for `tcpdump`) at node S (*Snd* for sender, *Rcv* for receiver) and,

per-packet timestamps T_X^S from which jitters J_X can be computed at the receiver.

The remainder of this section describes the design of our experiments to evaluate the impact of OML instrumentation on the performance and accuracy of applications, and on the system resources.

Application Performance

The first potential impact we want to test is the effect of OML on an application's performance. In the case of Iperf, we are interested in how the traffic generator performance and the accuracy of the reported measurements may be altered. Our experiment's null hypothesis is that the OML instrumentation has no significant impact on the packet sending rate of Iperf's traffic generator, and on the accuracy of Iperf's report of measured throughput and jitter.

Experimental Factors Four fixed factors may prove relevant to our working hypothesis above.

Iperf flavour and reporting mode The main concern for this experiment, we want to assess whether the OML instrumentation and the various reporting modes introduce a deviation from the standard version. This could be seen as two nested factors, *i.e.*, OML instrumentation or not, with the various reporting modes nested within. However, it made for a simpler design to flatten them into a single factor with four treatments:

nooml-c No modification, CSV reporting to a file;

oml-c OML instrumentation in the code, but CSV reporting to a file; this doubles as a sanity-check to make sure the instrumentation didn't break any mechanism;

oml-o OML instrumentation with legacy OML reporting;

oml-O OML instrumentation with advanced OML reporting.

Required sending rate At high rates the processing of each packet for measurement reporting may take longer than the inter-packet sending period, thus impending the sending rate. Although the sending rate is continuous, for simplicity, we treat it as a fixed factor (**1**, **10**, **50**, **95** and **100** Mbps) and do not focus on other values.

Threads One of Iperf's features is its use of threads (optional, but enabled by default) to report the measurements out of the main high-speed traffic generation loop. In the context of our study, this could hide performance impact introduced by the instrumentation. We therefore considered this factor with two treatments, **threads** and **nothreads**.

Transport protocol Vanilla Iperf supports both TCP and UDP.

Dependent Variables This experiment focuses on three dependent variables which we measure to characterise the potential influence of the OML instrumentation.

Actual sending rate Iperf's real sending rate as computed from `tcpdump` traces at the sender, R_t^{Snd}.

Accuracy of the throughput report The difference between the throughput reported by Iperf and the one measured by `tcpdump` at the receiver. For a given sample, $R_{\mathrm{Diff}}^{Rcv} = |R_t^{Rcv} - R_i^{Snd}|$.

Accuracy of the jitter report Similarly, the difference between the jitter reported by Iperf and the one computed from `tcpdump` at the receiver, $J_{\mathrm{Diff}} = |J_t - J_i|$, where J_t is computed from `tcpdump` traces as per (2.1) at the last packet of the period of each sample J_i.

Experiment Design Our final design is shown in Table 5.3. In this design, we only use UDP as the transport protocol, as it is unclear what proportion of the effect on the dependent variables would be due to the OML instrumentation or the TCP congestion control mechanisms. Furthermore, we decided not to consider the required sending rate as a direct factor, but rather run separate trials at each rate and consider their grouped results separately. Indeed, as the sending rate is one of our dependent variables, it is obvious that changing its set point will have an impact on this observed variable. Another approach would have been to normalise the measured rates with respect to this set point, but the first experiments showed a clear effect which may have belittled other factors of more interest to us.

In each trial, the measurement applications run for 300 s. For each sampling period of 0.5 s, the measured or computed dependent variables are reported. As we only focus on UDP

Table 5.3 Experiment design to characterise the impact of the OML instrumentation on Iperf.

nooml-c		**oml-c**	
threads	**nothreads**	**threads**	**nothreads**
598 samples	···	···	598 samples
oml-o		**oml-O**	
threads	**nothreads**	**threads**	**nothreads**
598 samples	···	···	598 samples

traffic, there is no transient adaptation period as is usually the case with TCP's slow-start. We however ignore the first and last sampling period of each trial as their boundaries do not match the application's starting/stopping times, thus resulting in incomplete data on the observed variables within that period.

In an earlier pilot study with a vanilla Iperf, we did not find any statistical difference between subsequent trials of the same experiment. We therefore assume that having 598 samples from one trial is equivalent to having one sample from 598 trials, thus ensuring suitable replication.

Packet Capture

To evaluate the effect of OML instrumentation on packet capturing, we use a similar setup as above, but with an experiment design involving only two dependent variables. In this case, our experiment's null hypothesis is that the OML instrumentation has no significant impact on the accuracy of packet capturing in terms of the number of observed packets and their timestamps.

Iperf is used to generate traffic at various rates between the two nodes. On each side, a plain `tcpdump` is used as a *reference* (t) to collect packet identifiers (from the IP header) and timestamps from the generated traffic. In addition on each side, another packet capture application is used as an *alternative* (a) to measure the same variables. This alternative can either be our OML instrumented `oml2_trace` application or another `tcpdump` instance.

Experimental Factors We consider two fixed factors in this experiment. The first one is the sending rate, with the same treatments as before: **1**, **10**, **50**, **95** and **100** Mbps. The second one is the use of the OML instrumented *alternative* application, or not. This corresponds to the two treatments **trace** (when an OML instrumented application is used), and **notrace**.

Dependent Variables This experiment focuses on two dependent variables.

Accuracy of packet report Missing packet reports may introduce a bias in the observations of a given research study. This may happen, *e.g.*, when the control network is saturated with large number of reports, resulting in retransmissions and losses of samples. Thus for each trial, we count the number of packets sent N_{sent} based on the IP identifiers from t, and the number l_a^S of packets not captured by a on side S, as identified by gaps in the sequence numbers; we then compute the loss ratio $L_a^S = l_a^S / N_{\text{sent}}$.

Accuracy of timestamp The difference in the reported timestamps between t and a on either node, $T_{\text{Diff}}^S = |T_t^S - T_i^S|$.

Experiment Design The timestamp precision T_{Diff}^S can be estimated for each packets in a trial. We therefore have access to many more samples of that variable than in the previous experiment (*e.g.*, of the order of 1×10^7 at 50 Mbps). In contrast, the number of unreported packets, N_{Loss}, is a single aggregate value for each trial. We therefore run 25 trials of this experiment to have sufficient replication and an acceptable number of samples of N_{Loss}.

System Resources

To evaluate the impact of OML instrumentation on system resources, for each of the treatment groups in the above experiments we also run the `oml2_nmetrics` application to monitor resource usage such as non-idle CPU time C^S and used memory M^S. In this context, our null hypothesis is that the OML instrumentation has no significant impact on the CPU and memory usage of the system running it.

Dependent Variables The following additional two variables are observed for each previously described experiments. We remove the trend displayed by both C^S and M^S by differentiating subsequent samples for each time interval $[T-1, T]$.

Differentiated CPU time $\Delta C^S = C_T^S - C_{T-1}^S$ and,

Differentiated memory usage $\Delta M^S = M_T^S - M_{T-1}^S$.

Experiment Design The experimental factors in this last experiment are the same as the ones for the previous experiments. We run `oml2_nmetrics` in each treatment groups. As it is an instrumented tool itself, we assumed it introduces a uniform bias to the system load for each trial. We therefore ensure that no further trend biases our dependent variables by starting `oml2_nmetrics` first to let it measure the idle system before actually starting the other instrumented tools. This doubles the previously mentioned trial duration. Thus with a sampling period of 5 s, we collect a total of 120 samples per node, but only the last 60 are representative of the impact of the other measurement tools.

Technical Details of the Experiments

The experiments described before were conducted on an OMF-enabled testbed. The experimental nodes *Snd* and *Rcv* are VIA MB770, 1 GHz CPU with 1 GB of RAM and two 100 Mbps Ethernet cards. The measurement server o is a 6-core AMD Phenom II

X6 1055T Processor with 12 GB of RAM, and is connected to the experiment nodes on a 1 Gbps internet LAN. The experiment nodes run Ubuntu Linux with kernel 2.6.35-25-generic #44-Ubuntu SMP.

5.4 Results

This section presents the results of the analyses which we performed on the collected experimental data.

Prior to analysing the data, we first assess the precision of our measurements by computing the relative standard error (RSE) of all measured variables within a trial for each treatment groups of the previously described experiments. The minimum and maximum RSE in our datasets are 8×10^{-6} % and 0.17 %. Thus our collected set of data has a sufficient precision (< 5 %) to be used in further analyses.

We now consider the dependent variables introduced in Section 5.3.2 and attempt to disprove our null hypotheses by identifying significant variations between each treatment groups in our experimental factors. To this end, we perform ANOVAs on the dependent variables. We choose the significance level $\alpha = 0.05$ to have 95 % confidence when finding significant differences.[7] We note that some of the assumptions of the ANOVA are not always met by our datasets. We address these problems as follows.

Independence of samples As the samples of each variables for one trial come from a time series, they are clearly not independent, which is confirmed by Turning points tests (Morley and Adams, 1989). We therefore make our data iid by sampling it randomly with replacement as suggested by Le Boudec (2010; sec. 2.3.3).

Homoskedasticity In some cases, Breusch-Pagan tests show that the variance of our samples differs significantly between treatment groups. Studies have however shown that the ANOVA is robust to deviations from this assumption at the price of a small reduction of the confidence $1 - \alpha$ and an increase of the power of the test β (Glass *et al.*, 1972; Harwell *et al.*, 1992). Moreover, we note that these studies focused on ratio of variances as low as 1:2. Even in our extreme cases, computing the ratio of the variances reveals that the heteroskedasticity is much more modest than the cases studied in those articles. We therefore conclude that our performed ANOVAs give us valid results even with this caveat on the confidence.

Normality This is the assumption from which our data deviated the most, both in terms of skewness and kurtosis (flatness). We characterised this deviation with a Shapiro-Wilk test for each treatment group and proceeded with an ANOVA if the deviation was not found to be significant. In the case that the deviation was significant, we used a non-parametric version of the ANOVA (PERMANOVA) which removes the assumption

[7] If a factor is found to have a significant impact, the probability of it being a false positive (Type I error) is smaller than 5 %.

on the source distribution of the data by creating an empirical null distribution through permutations of the samples throughout treatments (Anderson, 2001).

5.4.1 Application Performance

We performed two-way ANOVAs with interactions for each of the variables R_t^{Snd}, R_{Diff}^{Rcv} and J_{Diff} for each of the studied set rates. However, we judged the results for 100 Mbps invalid as the sender was rate-limited by its local network interface in all treatment groups. Characteristic results for other rates are presented below.

Actual Sending Rate

The results of the PERMANOVA for Iperf's sending rate as measured by an un-instrumented `tcpdump`, R_t^{Snd}, at rates 1, 50 and 95 Mbps are shown in Table 5.4. The results of the analysis for set rate 10 are similar to those for 50.

Table 5.4 Two-way PERMANOVA with interactions on the actual sending rate of Iperf, R_t^{Snd}.

	d.f.	SS	MS	F	p	Signif.
			1 Mbps			
oml	3	2.00×10^{6}	6.65×10^{5}	1.01	0.394	—
threads	1	4.62×10^{5}	4.62×10^{5}	0.70	0.418	—
oml:threads	3	4.89×10^{5}	1.63×10^{5}	0.25	0.864	—
Residuals	3192	2.11×10^{9}	6.61×10^{5}			
			50 Mbps			
oml	3	3.55×10^{11}	1.18×10^{11}	22.70	0.001	⋆⋆⋆
threads	1	2.33×10^{11}	2.34×10^{11}	44.76	0.001	⋆⋆⋆
oml:threads	3	3.55×10^{11}	1.18×10^{11}	22.74	0.001	⋆⋆⋆
Residuals	3192	1.66×10^{13}	5.21×10^{9}			
			95 Mbps			
oml	3	6.29×10^{11}	2.10×10^{11}	8.60	0.001	⋆⋆⋆
threads	1	9.03×10^{11}	9.03×10^{11}	37.03	0.001	⋆⋆⋆
oml:threads	3	3.43×10^{11}	1.14×10^{11}	4.68	0.003	⋆⋆
Residuals	3192	7.79×10^{13}	2.44×10^{10}			

Significance level: ⋆ 0.05, ⋆⋆ 0.01, ⋆⋆⋆ 0.001

For rates 10 Mbps and higher, there are statistically significant differences in Iperf's sending rates, which are introduced by changes in both the use of threads and OML instrumentation, as well as the interaction of those two factors. When significant, this interaction has to be studied first, which we do in Figure 5.5 for case 95 Mbps. This figure shows the so-called graph of means, with each treatment of the oml factor on the x-axis and a connecting line linking means for the same treatment of the thread factor. It shows an interaction between the threads and oml factors. For the threads treatment, the oml factor does not appear to have an impact, while its oml-o treatment affects R_t^{Snd} in the nothreads treatment.

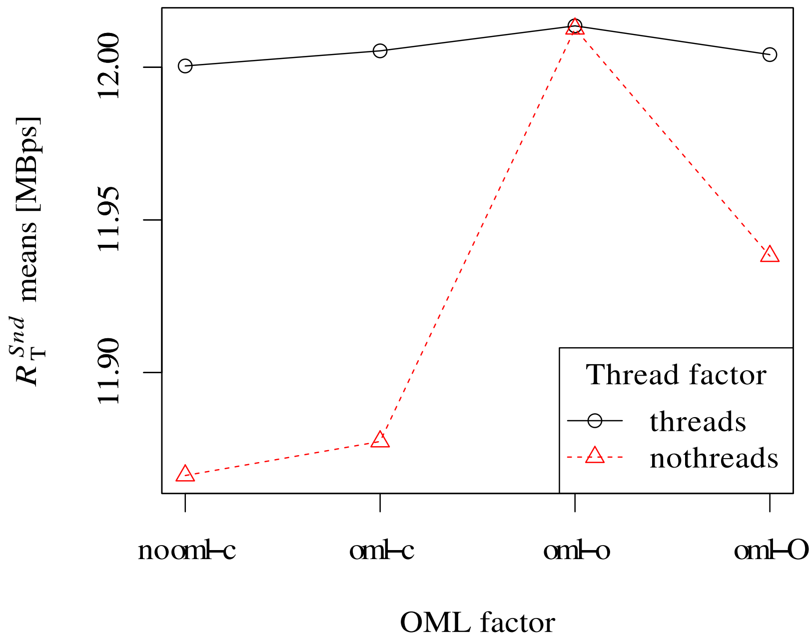


Figure 5.5 Graph of means for the interaction between experimental factors on R_t^{Snd} at 95 Mbps. Legacy OML reporting allows unthreaded Iperf to achieve a throughput with no significant difference from the threaded version (see Table 5.5).

The Tukey Honest Significant Differences test allows us to quantify the deviations observed in Figure 5.5. We present the relevant results allowing to characterise the previous figure in Table 5.5. For legibility's sake, we only show the mean differences and the p-values. We however include all the differences between interactions which were found to be significant.

Table 5.5 further indicates that, contrary to our expectations, the version integrating OML performs *better* than the vanilla version. Indeed, with a set rate of 95 Mbps, an unthreaded OML-instrumented Iperf was able, in the same conditions, to send an average of about 130 kBps ($\simeq$ 1 Mbps) more than the vanilla version reporting to a CSV file, which is the same as the threaded version. We found similar significant effects, though of smaller amplitude, at rates 10 Mbps and above.

Accuracy of the Throughput Report

Next, we assess the variations of R_{Diff}^{Rcv} between the treatment groups, as an evaluation of the impact of the instrumentation on the accuracy of Iperf's report. Table 5.6 presents the corresponding PERMANOVA results.

Table 5.5 Tukey Honest Significant Differences for oml:threads interactions with significant differences in the sending rate R_t^{Snd} at 95 Mbps. Only differences relevant to Figure 5.5 and all those found to be significant with 95% confidence are shown.

oml:threads interaction	diff [kBps]	p adj	Signif.
oml-c:nothreads–nooml-c:nothreads	−20.77	0.98	–
oml-o:nothreads–nooml-c:nothreads	107.41	0.00	⋆⋆⋆
nooml-c:threads–nooml-c:nothreads	88.27	0.00	⋆⋆⋆
oml-c:threads–nooml-c:nothreads	91.85	0.00	⋆⋆⋆
oml-o:threads–nooml-c:nothreads	108.71	0.00	⋆⋆⋆
oml-O:threads–nooml-c:nothreads	98.91	0.00	⋆⋆⋆
oml-o:nothreads–oml-c:nothreads	128.17	0.00	⋆⋆⋆
nooml-c:threads–oml-c:nothreads	109.04	0.00	⋆⋆⋆
oml-c:threads–oml-c:nothreads	112.62	0.00	⋆⋆⋆
oml-o:threads–oml-c:nothreads	129.47	0.00	⋆⋆⋆
oml-O:threads–oml-c:nothreads	119.67	0.00	⋆⋆⋆
oml-O:nothreads–oml-o:nothreads	−75.13	0.02	⋆
oml-o:threads–oml-o:nothreads	1.30	1.00	–
oml-o:threads–oml-O:nothreads	76.43	0.01	⋆⋆
oml-O:threads–oml-O:nothreads	66.64	0.05	⋆

Significance level: ⋆ 0.05, ⋆⋆ 0.01, ⋆⋆⋆ 0.001

Table 5.6 Two-way PERMANOVA with interactions on the difference R_{Diff}^{Rcv} between the actual received rate and Iperf's report.

	d.f.	SS	MS	F	p	Signif.
			1 Mbps			
oml	3	3.66×10^6	1.22×10^6	0.44	0.714	—
threads	1	9.62×10^6	9.62×10^6	3.47	0.061	—
oml:threads	3	3.26×10^5	1.09×10^5	0.04	0.990	—
Residuals	3192	8.85×10^9	2.77×10^6			
			50 Mbps			
oml	3	5.41×10^9	1.80×10^9	1.39	0.230	—
threads	1	1.42×10^{10}	1.42×10^{10}	10.96	0.001	⋆⋆⋆
oml:threads	3	6.05×10^9	2.02×10^9	1.56	0.191	—
Residuals	3192	4.14×10^{12}	1.30×10^9			
			95 Mbps			
oml	3	7.81×10^{15}	2.60×10^{15}	83300.57	0.001	⋆⋆⋆
threads	1	3.97×10^{12}	3.97×10^{12}	127.03	0.001	⋆⋆⋆
oml:threads	3	8.09×10^{12}	2.70×10^{12}	86.35	0.001	⋆⋆⋆
Residuals	3192	9.97×10^{13}	3.12×10^{10}			

Significance level: ⋆ 0.05, ⋆⋆ 0.01, ⋆⋆⋆ 0.001

At 1 and 50 Mbps, no statistically significant difference seems to be due to OML. At the latter rate, threads are the only source of deviation. At 95 Mbps, both factors, as well as their interaction, are found to have a significant effect. However, there is a discrepancy in the sum of squares for factor oml at this rate.

Further investigation of the data revealed that some packet reports in the advanced mode (oml-O treatment) are lost, thus introducing a bias in report accuracy similar to that mentioned in Section 5.3.2. This leads to underestimating (by 3–4 %) the throughput for that case when computing it *a posteriori* based on these reports. Because of the heavy bias introduced by this treatment, particularly on the homoskedasticity of the data, it is impossible to draw any conclusion from the analysis of this case.

To shed some light on the causes of these losses, we first verified that the problem did not arise because of a saturation of the control link. We found that, even at high rates (95 and 100 Mbps), the reporting generated a steady 3–4 Mbps, which is well below the 100 Mbps limit of the control network and the interfaces of the experiment nodes.

We also performed another ANOVA for this case, ignoring treatment oml-O. It showed no statistically significant difference ($p > 0.05$) between the other treatments of the oml factor, including OML legacy reporting. We then ran a modified oml-O trial where a summing filter was used at a 0.5 s interval, thus generating the same amount of reports to the server as in the oml-o treatment. An analysis of the data consisting of the previous set, with the reports from oml-O replaced by those from the newly introduced filtered trial suggests that no significant difference can be observed when using interval-based filters. However, as both approaches imply modifications of the experiment design, they cannot be rightly compared to the rest of the results presented here. We keep this as future work for the design of confirmatory experiments.

Based only on the remaining valid results we collected, we find no statistically significant difference in the accuracy of Iperf's throughput which is due to OML instrumentation or reporting. However, although no significant interaction effect is found at 50 Mbps, we still include the graph of means for R_{Diff}^{Rcv} in Figure 5.6. This graph shows a trend[8] which we find interestingly similar to that of Figure 5.5, thus suggesting that a similar significant interaction of factors may appear at 95 Mbps.

Accuracy of the Jitter Report

We finally attempt to find differences in J_{Diff} in a similar fashion. Due to the large amount of memory required by our implementation of the *a posteriori* jitter computation based on packet dumps, we could only study the cases in 1–50 Mbps. Table 5.7 summarises our findings, and shows a significant difference introduced by OML in the 10 Mbps ($p < .01$) and 50 Mbps ($p < .001$) cases.

We conduct Tukey HSD tests to evaluate this difference. They show that no statistically significant difference could be observed between vanilla Iperf with CSV output and OML legacy reporting. However, a significant increase in the jitter is observed in the case of the

[8]We do not refer to the apparent "trend" of the lines, which is a meaningless artifact of the graphical representation of categorical data. Rather, we mean the similarity of the behaviours observed for the same combination of factors.

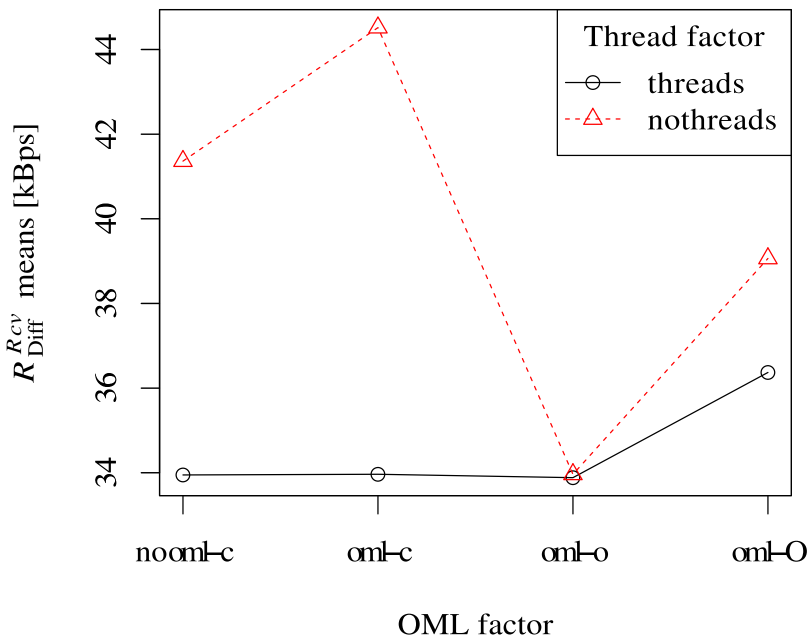


Figure 5.6 Graph of means for the interactions on Iperf's R_{Diff}^{Rcv} at 50 Mbps. OML reporting appears to increase the accuracy of the reports from the unthreaded trials, bringing them up to that of the threaded ones.

Table 5.7 Two-way PERMANOVA with interactions on the difference J_{Diff} between the actual jitter and Iperf's report.

	d.f.	SS	MS	F	p	**Signif.**
			1 Mbps			
oml	3	0.10	0.03	0.48	0.684	–
threads	1	0.00	0.00	0.05	0.833	–
oml:threads	3	0.25	0.08	1.22	0.322	–
Residuals	4784	322.15	0.07			
			10 Mbps			
oml	3	0.65	0.22	4.24	0.006	⋆⋆
threads	1	0.11	0.11	2.08	0.149	–
oml:threads	3	0.36	0.12	2.35	0.075	–
Residuals	4561	233.55	0.05			
			50 Mbps			
oml	3	1.85	0.62	51.18	0.001	⋆⋆⋆
threads	1	0.63	0.63	52.63	0.001	⋆⋆⋆
oml:threads	3	2.12	0.71	58.72	0.001	⋆⋆⋆
Residuals	4784	57.67	0.01			

Significance level: ⋆ 0.05, ⋆⋆ 0.01, ⋆⋆⋆ 0.001

advanced OML reporting at 50 Mbps, as illustrated by Figure 5.7. For the 10 Mbps case, the graph of means (not reported here) suggests a similar but much smaller effect. This effect size may be too modest to be detected with statistical significance by our current tests. Tests with higher statistical power may be able to detect such effect and will be considered for future work.

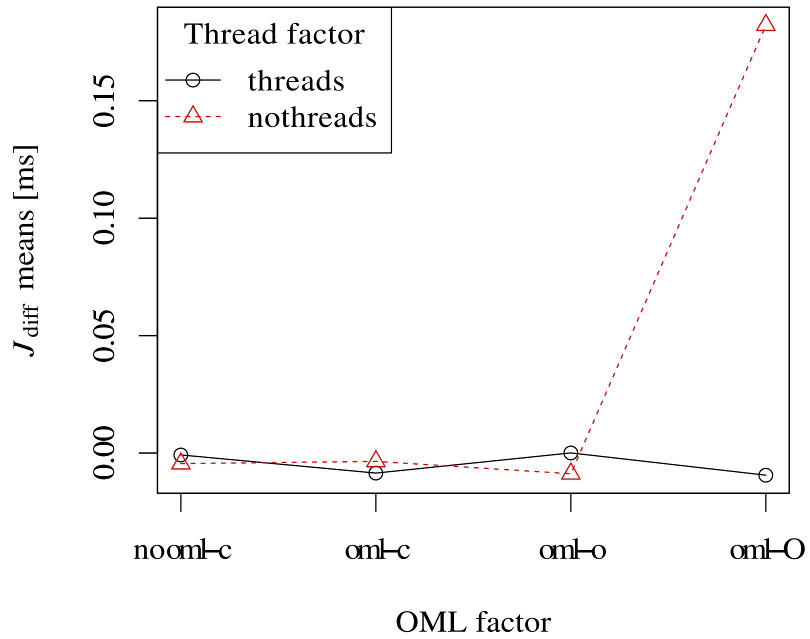


Figure 5.7 Graph of means for the interactions of experimental factors on J_{Diff} at 50 Mbps. Iperf's legacy OML reporting does not seem to introduce any significant effect.

5.4.2 Packet Capture

We performed a similar analysis on the relevant dependant variables of our packet-capture experiment. Characteristic results are reported thereafter.

Accuracy of Reports

We perform a two-way PERMANOVA on the loss ratio L_a^{Snd} for our treatment groups in the packet capture experiment. Table 5.8 presents a summary of this analysis. Though the trace factor seems to have the most statistically significant impact, its interaction with the rate factor needs to be studied first. The related graph of means in Figure 5.8 shows that the interaction of the trace factor and high rates has the largest effect, increasing the packet loss ratio.

In Table 5.9, we reproduce those results of the Tukey HSD test which are relevant to the interpretation of Figure 5.8. It shows that, at low rates (1–50 Mbps), the difference is not

Table 5.8 Two-way PERMANOVA with interactions on the loss ratio L_a^{Snd}.

	d.f.	SS	MS	F	p	**Signif.**
trace	1	0.07	0.07	9.24	0.001	⋆⋆⋆
rate	4	0.08	0.02	2.56	0.027	⋆
trace:rate	4	0.08	0.02	2.65	0.021	⋆
Residuals	200	1.48	0.01			

Significance level: ⋆ 0.05, ⋆⋆ 0.01, ⋆⋆⋆ 0.001

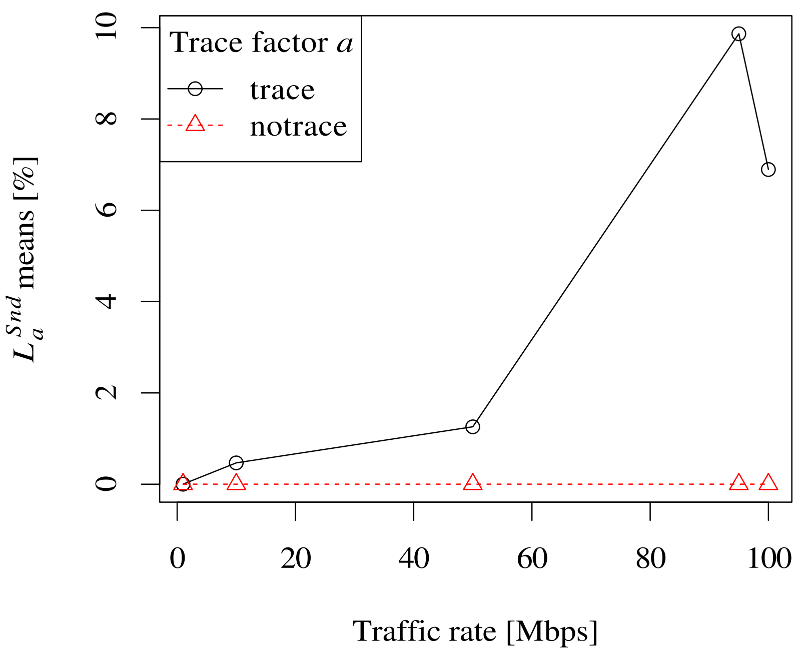


Figure 5.8 Graph of means for the interaction of factors trace and rate on the loss ratio L_a^{Snd}.

statistically significant but becomes significant for the 95 Mbps treatment. We however note that, regardless of significance, the difference is always positive, which means that our OML-instrumented libtrace tool tends to miss more packets than `tcpdump`.

Timestamp Accuracy and Precision

As mentioned in section 5.3.1, the timestamps for both tools come directly from the kernel capture. We found the accuracy for both the reference and the alternative to always be equal ($T^S_{\text{Diff}} = 0$).

It is worthwhile to note that, while both libraries can access and store packet timestamp information with a microsecond resolution, we found in our experiments that the libpcap trace files only had a millisecond resolution. We therefore discarded the microsecond information from `oml2_trace` when computing T^S_{Diff}.

Table 5.9 Tukey HSD for trace:rate interactions with significant differences in the loss ratio L_a^{Snd}. Only differences relevant to Figure 5.8 are shown.

trace:rate	diff [%]	p adj	Signif.
trace:1-notrace:1	9.08×10^{-4}	1.00	–
trace:10-notrace:10	0.46	1.00	–
trace:50-notrace:50	1.26	1.00	–
trace:95-notrace:95	9.87	0.02	$\star$
trace:100-notrace:100	6.89	0.26	–

Significance level: $\star$ 0.05, $\star\star$ 0.01, $\star\star\star$ 0.001

5.4.3 System Resources

We analysed the system usage variables ΔC^S and ΔM^S in the same manner as the other dependent variables, for both experiments on Iperf performance and packet capture. Some of the resulting ANOVAs showed significant effects. However, further investigation showed that in these cases the affected node had been doing memory maintenance tasks, thus confounding our results.

This leads us to conclude that, overall, no statistically significant deviation can be found, either in terms of CPU or memory usage, between the different treatment groups involving OML enabled tools or not.

5.5 Discussion

Based on the interpretation of the results just presented, we now discuss their implications with respect to the performance of OML and the studied applications. We then provide recommendations for the use of OML within our framework.

5.5.1 On Application Performance

Our study of an OML instrumented Iperf gave us insight, both on the influence of OML and Iperf itself, which we present here.

Vanilla Version

As a side effect of our OML impact study, we have performed what can be considered one of the first deep investigations of Iperf's performance and reporting accuracy. This performance analysis of Iperf brought some insight on its potential impact on networking research. In particular, we unexpectedly found that the Iperf reporting function can impact the accuracy and performance of the traffic generation up to 2% in our worst case scenario. This difference can be explained by the reporting functions, which are run synchronously in the non-threaded version of vanilla Iperf.

This discovery raises several questions about previously published Iperf-based measurements. Indeed, although most distributions provide an Iperf built with threads support, some platforms—mostly embedded—do not support threads (*e.g.*, Windows CE Pocket PCs).

OML-Instrumented Flavour

Nevertheless, the analysis of the impact of legacy OML reporting on unthreaded Iperf suggests that, even if the reporting is done within the main thread, the use of OML instead of the usual CSV file-writer allows to reach similar performances as the threaded builds. It is important to note that the OML reporting did not introduce any threads in this case.

Our results also show that reporting every packet information for Iperf (advanced mode) could impact the overall performance of both threaded and unthreaded versions of Iperf. Early results from confirmatory experiments however suggest that the use of OML filters before reporting measurements can mitigate this issue. This result is consistent with that of the packet capturing tool as discussed next.

5.5.2 On Reporting Accuracy

Our results for the packet capture software show that there is a significant difference between our OML-instrumented wrapper for libtrace and `tcpdump` at high rates. In our worst case scenario the OML based tool would lose an average of about 10 % more packets at 95 Mbps. This behaviour may be explained by two main reasons.

First, the OML architecture may contain a bottleneck which would cause measurements to be lost when generated at too high a rate. Indeed, as illustrated in Figure 5.9, when a sample is injected by the client, the data stream is transferred reliably to the OML server over TCP. On the server side, the samples are inserted into an SQLite database. The server's default parameter for OML is to perform one writing transaction every second. In this study, two clients were reporting two MPs (IP and TCP headers) per packet. This resulted in four injections on the server side. At high rates, too many packets may therefore have been reported for the OML server to be able to empty the TCP buffers in time. In those cases, the TCP receiver would start sending zero-window advertisements, therefore blocking the TCP sender and, finally, the injection function. As our basic capturing packet application was not threaded, blocking the measurement injection in turns caused packet capture buffers to be overwritten before being read by the application. The same hypothesis can be proposed to explain the relatively poorer performance of the oml-O treatment of our Iperf experiment at high rates.

Also, in `oml2_trace` we do not dump the binary packet directly into a file as `tcpdump` does, but rather access specific fields of the relevant headers. This process is done for every packet prior to the injection into the measurement stream. We therefore suspect our application may be slowed down even before the OML process is started. In order to have a fairer comparison, it would be interesting to conduct a complementary experiment. The notrace treatment would either be `tcpdump` operated in a verbose mode and dumping out

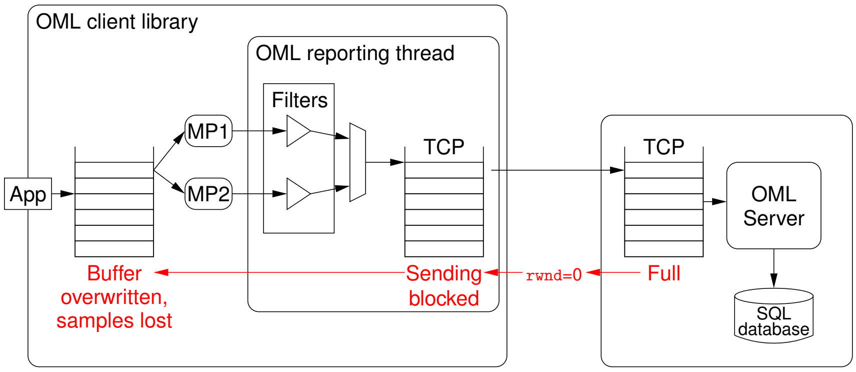


Figure 5.9 Data path from injection to database. When the transaction takes place on the server side, the TCP receiver buffer fills up. As a result a null `rwnd` is advertised to the client, which results at the other end in data discarding in the application.

the parsed packets in a file, or `oml2_trace` configured to directly dump the samples from the captured packets to a local storage.

5.5.3 On Resource Usage

For all experiments, the OML instrumentation has not been found to induce any significant impact on the system's CPU and memory usage. These results were expected for the experiments at low traffic rates but not for the ones at higher rates. In particular we anticipated that the advanced OML reporting mode of Iperf would show an increased system load, but our results did not identify such impact. Overall, our results suggest that the resource footprint of OML can be considered as negligible.

5.5.4 Recommendations for OML Instrumentation

We claim that OML is a good candidate as the information-collection bus for our cross-layer framework. Indeed, aside from its design goal of centralising measurement collection from multiple sources and storing them in a unified format, our analysis suggests that, in most cases, this library does not have a significant impact on the instrumented applications or the host system. It also simplifies the development of active measurement applications by removing the need for a complex threaded architecture. Nevertheless some considerations have to be taken into account when instrumenting applications.

A developer, be they in charge of writing a new tool or just instrumenting already existing applications, must be careful in establishing which metrics should be reported together in one MP. It is a trade-off between providing flexibility to the system and limiting the number of MPs—and, ultimately, generated samples—that need be reported to the server. For example, it may be relevant to group fields for which new measurements always arrive at the same time (such as various metrics for a packet). However, only some aspects may ultimately be of interest to the experimenter (*e.g.*, throughput but not jitter). Thus MPs with a lot of fields may only result in wasted database storage space.

Then, an experimenter would have to be careful with respect to the number of MPs selected to be reported to the OML server, and make sure it does not receive more than a few reports per millisecond. To do so one should take advantage of the filtering capabilities of OML to pre-process data on the client side and reduce the number of samples.

5.6 Conclusion and Future Work

We have conducted an in-depth performance evaluation of the OML 2.5.0. With a reasonable sample rate, we could not find any significant negative impact of such an instrumentation. Moreover, we identified statistically significant positive effects in some cases where the instrumented application did not use threads. Indeed, OML removes the complexity of reporting from an application's main purpose, therefore making it an interesting and unintrusive tool to collect metrics from applications which main purpose is not measurement. However, our results indicate that a single OML server cannot currently scale to sample rates of the order of more than a few per millisecond, though preliminary experiments suggest that this issue can be avoided by the use of OML filters.

Incidentally to this evaluation, we could also identify a shortcoming in the performance of Iperf 2. We found that disabling threads on the vanilla version has a significant impact on the accuracy of the generated rate and measurements. This raises concerns about the validity of Iperf-based studies comparing platforms which do not support threads with others which do.

Finally, we presented a set of recommendations for software instrumentation with OML and use as part of our proposed framework. We believe that, with a proper use of filtering before aggregating the reported metrics in the database, OML is a valid choice as the information-passing bus of our cross-layer framework. We intend to instrument more applications for integration in our proposed system. However, the operation of OML's filters should be studied more thoroughly, to the same level of what we did in this chapter, to formally confirm this hypothesis.

Chapter 6

Conclusion and Future Work

6.1 Research Challenges

The evolution of mobile technologies, including both communications and devices, is creating a number of challenges for the classical protocols used in the Internet, which were designed for fixed, wired networks. The increased use of wireless connectivity, together with a growing number of wireless technologies and networks which may be used simultaneously by a mobile device, create both opportunities to be always connected to a network which will provide the best quality, and performance issues when mobility events occur. In addition, more powerful mobile devices and emerging, more demanding, applications put additional strain on the required performance and the supporting networks and protocols.

More specifically, important issues which need to be addressed include

- appropriate network selection so that the requirements of applications, devices and users can be catered for to the best within the limits of the current available network, and
- provision of new functionality in the network stack so that it can adapt to dynamically changing conditions due to mobility and heterogeneous wireless technologies.

The contributions presented in this dissertation address these problems. They are related in the next section.

6.2 Contributions of this Thesis

To address these issues, we have introduced a cross-layer control framework for the protocol stack of a mobile node. The framework consists of three classes of elements. The first one is the *monitoring* of the current environment, in terms of running applications, state of the device, usable access networks and other contextual information. Based on this information, a *decision system* derives the parameters for all the stack elements (*e.g.*, network selection, transport protocols' or applications' parameters) which globally provide an appropriate performance. To act based on this decision, parameters of some of the elements of the stack need to be updated. This necessitates the development of *new interfaces and algorithms*. The contributions of this thesis aim at providing proposals for those three elements.

Our first contribution is a generic model of a multihomed mobile device, its network environment and the currently running applications. We showed, using constrained optimisation, that directly considering high level metrics allows for improved performance quality. We have argued that considering user-perceivable criteria such as QoE, battery lifetime or price is a better approach than considering QoS- and other network metrics-based criteria. We concluded that these user- and application-centric criteria should therefore be directly considered in network selection and flow distribution mechanisms.

As our second contribution, we introduced a mobility-aware extension to the TCP-Friendly Rate Control (TFRC), to address the disruption in connectivity resulting from the mobility hand-offs and flow redistribution. Using simulations, we demonstrated how this enables faster recovery after disconnected periods as well as significantly improved adjustments to the newly available network conditions. We showed that, when used with the Datagram Congestion Control Protocol (DCCP), the proposed mechanism provides improved support for real-time applications, for which the user-perceived quality is highly dependent on the immediate transmission rate. We have also demonstrated, based on our Linux implementation, that this proposal can provide a mobile video-streaming application with a higher quality throughout several changes of supporting networks, compared to what can be achieved with standard TFRC.

Our third contribution is a performance evaluation of the OMF Measurement Library (OML) for reporting and collection of measurements. This library is a lightweight off-the-shelf reporting tool which can be used to instrument third-party applications, allowing for a unified monitoring bus across all of the stack. We therefore proposed to use OML as the observation element of our framework. To justify our choice, we showed that OML does not significantly impact the performance of the instrumented applications, while accurately reporting the observed metrics.

6.3 Future Work and Perspectives

This thesis highlighted and evaluated elements and mechanisms for various steps of an adaptation system to better exploit the network resources available to a mobile node. However, this has been done separately for each part. The obvious and most needed next step is to integrate the contributions from this thesis into a coherent system following the described architecture.

The following task would be to globally evaluate this system. Performance or quality improvement from the perspective of the device running our cross-layer framework is the main evaluation metric. However, it is also important to ensure large scale use of our proposal does not introduce instabilities in the patterns of access network usage or, if so, to study and integrate solutions to these instabilities.

Additionally, the work presented here was working under the strong assumption that all elements of our proposed framework were colocated within the same device. With the advent of NEMOs, this assumption is violated. Extension of our proposal to distributed systems and decision, where applications run on devices different from the routers, is a desirable extension. This would be particularly appropriate for ITS applications implementing the ISO/ETSI ITS Station reference architecture ETSI EN 302 665 (2010), which already provides a set of cross-layer functions for interface and path selection.

From a network stack perspective, it is evident that the transport layer has taken upon itself to serve many roles and provide many features missing from the lower layers (*e.g.*, session control or path management). While such features truly belong at that layer, it may serve the whole architecture better if some functionalities were relocated to more appropriate layers. For example, a richer network layer could identify all paths to a given host, and provide this information in a standard way to the transport protocol, along with unified congestion information. The upper layer would only be left in charge of its original tasks, such as connected mode semantics or reliability, while per-node congestion control would be done only once for each path. This would also greatly simplify the development of multipath-aware transport, by introducing a clearer understanding of what a path is, and leaving only adequate packet scheduling to the transport.

Throughout work on this thesis, it also became apparent that one of the main limitations in the migration path from legacy protocols of the TCP/IP stack was the socket API. Indeed as it over-specifies the requirements an application has on the protocol stack, it prevents smooth migration to newer alternative providing the same services without intervention of the application developer (*e.g.*, from IPv4 to IPv6, or from TCP to a more adapted transport protocol). The current API also ignores newer, but now common, parameters (*e.g.*, firewall configuration), leaving it up to the application itself to perform such ancillary tasks which should be globally provided by the system. While the low level socket API provides a tested foundation to establishing network connections, perhaps it is time to revisit the interface which is exposed to high-level networked applications.

Bibliography

3GPP TS 22.101. Service aspects; service principles. 3GPP/TSG SA1, Release 11.0.0, September 2010. 1

3GPP TS 25.201. Universal mobile telecommunications system (UMTS); physical layer — general description. 3GPP/TSG R WG1, Release 10.0.0; Also published as ETSI TS 125 201, April 2011. 16, 17

3GPP TS 25.306. UE radio access capabilities. 3GPP/TSG RAN2, Release 10.1.0, December 2010. 16, 17

3GPP TS 29.275. Proxy mobile IPv6 (PMIPv6) based mobility and tunnelling protocols; stage 3. 3GPP/TSG CT2, Release 10.0.0, December 2010. 20

Evgenia Adamopoulou, Konstantinos Demestichas, Artemis Koutsorodi, and Michael Theologou. Intelligent access network selection in heterogeneous networks - simulation results. In Mario Gerla and Maria L. Merani, editors, *ISWCS 2005, 2nd International Symposium on Wireless Communications Systems*, pages 279–283. University of Siena, IEEE Communications Society, September 2005. ISBN 0-7803-9206-X. doi: 10.1109/ISWCS.2005.1547704. 33, 34

Ian Akyildiz, Shantidev Mohanty, and Jiang Xie. A ubiquitous mobile communication architecture for next-generation heterogeneous wireless systems. *IEEE Communications Magazine*, 43(6):S29–S36, June 2005. ISSN 0163-6804. doi: 10.1109/MCOM.2005.1452832. 33, 34

Ian F. Akyildiz, Jiang Xie, and Santidev Mohanty. A survey of mobility management in next-generation all-IP-based wireless systems. *IEEE Wireless Communications*, 11(4):16–28, August 2004. ISSN 1536-1284. doi: 10.1109/MWC.2004.1325888. 20

Shane Alcock, Perry Lorier, and Richard Nelson. Libtrace: A trace capture and processing library. Technical report, University of Waikato, May 2010. 43, 104, 106

Mohammed Alkhawlani and Aladdin Ayesh. Access network selection based on fuzzy logic and genetic algorithms. *Advances in Artificial Intelligence*, 2008, 2008. doi: 10.1155/2008/793058. 33, 34, 35, 47

Mark Allman, Vern Paxson, and W. Richard Stevens. TCP congestion control. RFC 2581, RFC Editor, April 1999. 15

Mark Allman, Vern Paxson, and Ethan Blanton. TCP congestion control. RFC 5681, RFC Editor, September 2009. 14

Guy Almes, Sunil Kalidindi, and Matthew J. Zekauskas. A one-way delay metric for IPPM. RFC 2679, RFC Editor, September 1999a. 39, 53

Guy Almes, Sunil Kalidindi, and Matthew J. Zekauskas. A one-way packet loss metric for IPPM. RFC 2680, RFC Editor, September 1999b. 39

Guy Almes, Sunil Kalidindi, and Matthew J. Zekauskas. A round-trip delay metric for IPPM. RFC 2681, RFC Editor, September 1999c. 39

Timur Alperovich and Brian Noble. The case for elastic access. In Arun Venkataramani and Marco Gruteser, editors, *MobiArch 2010, 5th ACM International Workshop on Mobility in the Evolving Internet Architecture*. ACM SIGMOBILE, ACM, September 2010. ISBN 978-1-4503-0143-5. 33

Marti J. Anderson. A new method for non-parametric multivariate analysis of variance. *Austral Ecology*, 26(1):32–46, February 2001. doi: 10.1111/j.1442-9993.2001.01070.pp.x. 115, 189

ANSI T1.TR.74-2001. Objective video quality measurement using a peak-signal-to-noise-ratio (PSNR) full reference technique. Technical Report T1.TR.74-2001, American National Standards Institute, Ad Hoc Group on Video Quality Metrics, 2001. 41, 189

Jari Arkko, Christian Vogt, and Wassim Haddad. Enhanced route optimization for mobile IPv6. RFC 4866, RFC Editor, May 2007. 21

Basharat Ashai, Cintia Garza, and Ann Howe. 4GCounts quarterly report — Q3 2011. Report 16, Maravedis, November 2011. 17

Stefan Aust, Carmelita Goerg, and Cornel Pampu. Policy based mobile IPv6 handover decision (POLIMAND). Internet-Draft draft-iponair-dna-polimand-02.txt, IETF Secretariat, February 2005. 34

Stefano Avallone, Salvatore Guadagno, Donato Emma, Antonio Pescapè, and Giorgio Ventre. D-ITG distributed internet traffic generator. In Giuliana Franceschinis, Joost-Pieter Katoen, and Murray Woodside, editors, *QUEST 2004, 1st International Conference on Quantitative Evaluation of Systems*, pages 316–317. University of Twente, IEEE Computer Society, September 2004. doi: 10.1109/QEST.2004.1348045. 43

Adeel Baig, Lavy Libman, and Mabhub Hassan. Performance enhancement of on-board communication networks using outage prediction. *IEEE Journal on Selected Areas in Communications*, 24(9):1692–1701, September 2006. ISSN 0733-8716. doi: 10.1109/JSAC.2006.875108. 27

Ajay Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. In Jane Liu, editor, *ICDCS 1995, 15th International Conference on Distributed Computing Systems*, pages 136–143. IEEE Computer Sciety, March 1995. ISBN 0-8186-7025-8. doi: 10.1109/ICDCS.1995.500012. 20

Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, and Randy H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, December 1997. ISSN 1063-6692. doi: 10.1109/90.650137. 18

Hari Balakrishnan, Hariharan S. Rahul, and Srinivasan Seshan. An integrated congestion management architecture for Internet hosts. *SIGCOMM Computer Communication Review*, 29(4):175–187, October 1999. ISSN 0146-4833. doi: 10.1145/316194.316220. 29

Roberto Baldessari, Wenhui Zhang, Andreas Festag, and Long Le. A MANET-centric solution for the application of NEMO in VANET using geographic routing. In Miguel P. de Leon, editor, *TridentCom 2008, 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, pages 1–7. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), March 2008. ISBN 978-963-9799-24-0. 13

Deepak Bansal, Hari Balakrishnan, Sally Floyd, and Scott Shenker. Dynamic behavior of slowly-responsive congestion control algorithms. In Rene Cruz and George Varghese, editors, *SIGCOMM 2001, Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 263–274. ACM, August 2001. ISBN 1-58113-411-8. doi: 10.1145/383059.383080. xviii, 5, 8, 14, 18, 28, 94

Farooq Bari and Victor Leung. Automated network selection in a heterogeneous wireless network environment. *IEEE Network*, 21(1):34–40, January 2007. ISSN 0890-8044. doi: 10.1109/MNET.2007.314536. 33, 35

Rayene Ben Rayana and Jean-Marie Bonnin. Mobility aware application manager for mobile networks. In Passakon Prathombutr, Shozo Komaki, Gabrielle Landrac, Wasan Pattara-atikom, and GuangJun Wen, editors, *ITST 2008, 8th International Conference on Intelligent Transport Systems Telecommunications*, pages 337–342. IEEE Computer Society, October 2008. ISBN 978-1-4244-2857-1. doi: 10.1109/ITST.2008.4740282. 29, 72

Konstantinos Benekos, Nikos Pogkas, Grigorios Kalivas, George Papadopoulos, and Anthony Tzes. TCP performance measurements in IEEE 802.11b-based wireless LANs. In Maja Matijasevic, Branimir Pejcinovic, Zeljko Tomsic, and Zeljko Butkovic, editors, *MELECON 2004. 12th IEEE Mediterranean Electrotechnical Electrotechnical Conference*, volume 2, pages 575–578. University of Zagreb, IEEE Computer Society, May 2004. ISBN 0-7803-8271-4. 16, 18

Saâd Biaz and Xia Wang. Can ECN be used to differentate congestion losses from wireless losses? Technical Report CSSE04-04, Auburn University, May 2004. 27

Jean-Marie Bonnin. *La diversité technologique au service des terminaux et routeurs multiconnectés*. Mémoire d'habilitation à diriger des recherches, Institut Télécom—Télécom Bretagne, 2008. 33, 35

Jean-Marie Bonnin, Imed Lassoued, and Zied Ben Hamouda. Automatic multi-interface management through profile handling. *Mobile Networks and Applications*, 14(1):4–17, February 2009. ISSN 1383-469X. doi: 10.1007/s11036-008-0117-6. 35

Alessio Botta, Alberto Dainotti, and Antonio Pescapè. Multi-protocol and multi-platform traffic generation and measurement. In Jennifer Hou, editor, *INFOCOM 2007, 26th IEEE International Conference on Computer Communications, Demonstration Session*. IEEE Communications Society, May 2007. 43

Laurent Bouraoui, Stéphane Petti, Anis Laouiti, Thierry Fraichard, and Michel Parent. Cybercar cooperation for safe intersections. In Urbano Nunes, editor, *ITSC 2006, 9th International IEEE Conference on Intelligent Transportation Systems*, pages 456–461. IEEE Computer Society, September 2006. ISBN 1-4244-0093-7. doi: 10.1109/ITSC.2006.1706783. 4

John R. Boyd. The essence of winning and losing, June 1995. xvii, 6, 189

Robert Braden, Ted Faber, and Mark Handley. From protocol stack to protocol heap: Role-based architecture. *SIGCOMM Computer Communication Review*, 33(1):17–22, January 2003. ISSN 0146-4833. doi: 10.1145/774763.774765. 30

Lawrence S. Brakmo and Larry L. Peterson. TCP Vegas: End to end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8): 1465–1480, October 1995. ISSN 0733-8716. doi: 10.1109/49.464716. 17

Christof Brandauer and Thomas Fichtel. MINER — a measurement infrastructure for network research. In Thomas Magedanz and Shiwen Mao, editors, *TridentCom 2009, 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, pages 1–9. IEEE Computer Society, April 2009. ISBN 978-1-4244-2846-5. doi: 10.1109/TRIDENTCOM.2009.4976235. 45

Bob Briscoe. Flow rate fairness: Dismantling a religion. *SIGCOMM Computer Communication Review*, 37(2), April 2007. ISSN 0146-4833. doi: 10.1145/1232919.1232926. 14, 96

Kevin Brown and Suresh Singh. M-TCP: TCP for mobile cellular networks. *SIGCOMM Computer Communication Review*, 27(5):19–43, October 1997. ISSN 0146-4833. doi: 10.1145/269790.269794. 20

Lukasz Budzisz, Ramon Ferrús, Anna Brunström, Karl J. Grinnemo, R. Fracchia, G. Galante, and Ferran Casadevall. Towards transport-layer mobility: Evolution of SCTP multihoming. *Computer Communications*, 31(5):980–998, March 2008. ISSN 0140-3664. doi: 10.1016/j.comcom.2007.12.014. 20

Ian S. Burnett, Fernando Pereira, Rik Van de Walle, and Rob Koenen, editors. *The MPEG-21 Book*. John Wiley & Sons, June 2006. ISBN 978-0-470-01011-2. doi: 10.1002/0470010134. 28

Joseph Camp and Edward Knightly. Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation. In Raghupathy Sivakumar and Peter Steenkiste, editors, *MobiCom 2008, 14th ACM international conference on Mobile computing and networking*, pages 315–326. ACM SIGMOBILE, ACM, September 2008. ISBN 978-1-60558-096-8. doi: 10.1145/1409944.1409981. 24

Andrew Campbell, Javier Gomez, Chieh-Yih Wan, Sanghyo Kim, Zoltan R. Turanyi, and Andras Valko. Cellular IP. Internet-Draft draft-ietf-mobileip-cellularip-00.txt, IETF Secretariat, January 2000. 20

Casey Carter, Robin Kravets, and Jean Tourrilhes. Contact networking: A localized mobility system. In Mary Baker and Robert T. Morris, editors, *MobiSys 2003, 1st international conference on Mobile systems, applications and services*, pages 145–158. ACM SIGMOBILE, ACM, May 2003. doi: 10.1145/1066116.1066119. 29, 33

Pauline M. L. Chan, Ray E. Sheriff, Yim-Fun Hu, Paolo Conforto, and Clementina Tocci. Mobility management incorporating fuzzy logic for heterogeneous a IP environment. *IEEE Communications Magazine*, 39(12):42–51, December 2001. ISSN 0163-6804. doi: 10.1109/35.968811. 34, 35

Zhijiang Chang and Georgi Gaydadjiev. A hybrid cross layer architecture for wireless protocol stacks. In Xi Zhang, editor, *IWCMC 2008, International Wireless Communications and Mobile Computing Conference*, pages 279–285. IEEE, August 2008. ISBN 978-1-4244-2201-2. doi: 10.1109/IWCMC.2008.49. 29

Kai Chen, Klara Nahrstedt, and Nitin H. Vaidya. The utility of explicit rate-based flow control in mobile ad hoc networks. In Steve Weinstein, editor, *WCNC 2004, IEEE Wireless Communications and Networking Conference*, volume 3, pages 1921–1926. IEEE Computer Society, March 2004. ISBN 0-7803-8344-3. doi: 10.1109/WCNC.2004.1311847. 15

Mung Chiang. To layer or not to layer: Balancing transport and physical layers in wireless multihop networks. In Bo Li and Marwan Krunz, editors, *INFOCOM 2004, 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2525–2536. IEEE Computer Society, March 2004. ISBN 0-7803-8355-9. doi: 10.1109/INFCOM.2004.1354673. 28

Phil Chimento and Joseph Ishac. Defining network capacity. RFC 5136, RFC Editor, February 2008. 37, 38, 52

Johnny Choque, Ramón Agüero, Eva-María Hortigüela, and Luis Muñoz. Optimum selection of access networks within heterogeneous wireless environments based on linear programming techniques. In Marta García-Arranz and Symeon Papavasileiou, editors, *MONAMI 2010, Second International ICST Conference on Mobile Networks and Management*. Universidad de Cantabria, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), September 2010. 34

Song Ci, Haohong Wang, and Dalei Wu. A theoretical framework for quality-aware cross-layer optimized wireless multimedia communications. *Advances in MultiMedia*, 2008(2): 1–10, 2008. ISSN 1687-5680. doi: 10.1155/2008/543674. 28, 31, 32, 36

Benoit Claise, Stewart Bryant, Simon Leinen, Thomas Dietz, and Brian H. Trammell. Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information. RFC 5101, RFC Editor, January 2008. 44

David D. Clark. Toward the design of a Future Internet. October 2009. xvii, 5, 12

COMeSafety project. European ITS communication architecture — overall framework — proof of concept implementation. Deliverable COMeSafety-D.3.1-v2.0, EC Information Society Technologies Programme, March 2009. 3

Marco Conti, Silvia Giordano, Gaia Maselli, and Giovanni Turi. MobileMAN: Mobile metropolitan ad hoc networks. In Marco Conti, Silvia Giordano, Enrico Gregori, and Stephan Olariu, editors, *PWC 2003, 8th IFIP-TC6 International Conference on Personal Wireless Communications*, volume 2775/2003 of *Lecture Notes in Computer Science*, pages 169–174. IFIP, Springer-Verlag Berlin, September 2003. ISBN 3-540-20123-8. doi: 10.1007/b12004. 29

Marco Conti, Gaia Maselli, Giovanni Turi, and Silvia Giordano. Cross-layering in mobile ad hoc network design. *Computer*, 37(2):48–51, August 2004. ISSN 0018-9162. doi: 10.1109/MC.2004.1266295. 29

Daniel Corujo, Carlos Guimarães, Bruno Santos, and Rui L. Aguiar. Using an open-source IEEE 802.21 implementation for network-based localized mobility management. *IEEE Communications Magazine*, 49(9):114–123, September 2011. ISSN 0163-6804. doi: 10.1109/MCOM.2011.6011742. 30

Michelle Cotto and Leo Vegoda. Special use IPv4 addresses. RFC 5735, RFC Editor, January 2010. 2

Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In David B. Johnson, Anthony D. Joseph, and Nitin H. Vaidya, editors, *MobiCom 2003, 9th annual international conference on Mobile Computing and networking*, pages 134–146. ACM SIGMOBILE, ACM, September 2003. ISBN 1-58113-753-2. doi: 10.1145/938985.939000. 25, 188

Rina Dechter. *Constraint Processing*. Morgan Kaufmann Publishers, May 2003. ISBN 978-1-55860-890-0. 65, 187

Stephen E. Deering and Robert M. Hinden. Internet protocol, version 6 (IPv6) specification. RFC 2460, RFC Editor, December 1998. 2, 12

Carlo Demichelis and Philip Chimento. IP packet delay variation metric for IP performance metrics (IPPM). RFC 3393, RFC Editor, November 2002. 39

Vijay Devarapalli, Ryuji Wakikawa, Alexandru Petrescu, and Pascal Thubert. Network mobility (NEMO) basic support protocol. RFC 3963, RFC Editor, January 2005. 22, 189

Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In Samir R. Das and Ravi Jain, editors, *MobiCom 2004, 10th annual international conference on Mobile Computing and networking*, pages 114–128. ACM SIGMOBILE, ACM, October 2004. ISBN 1-58113-868-7. doi: 10.1145/1023720.1023732. 25, 188

Tom Dunigan, Matt Mathis, and Brian Tierney. A TCP tuning daemon. In Daniel Reed, editor, *SC 2002, ACM/IEEE conference on Supercomputing*, pages 9–25. IEEE

Computer Society, ACM SIGARCH, IEEE Computer Society, November 2002. ISBN 0-7695-1524-X. doi: 10.1109/SC.2002.10023. 29

Wesley Eddy. Mobility support for TCP. Internet-Draft draft-eddy-tcp-mobility-00.txt, IETF Secretariat, April 2004a. 20

Wesley M. Eddy. At what layer does mobility belong? *IEEE Communications Magazine*, 42(10):155–159, October 2004b. ISSN 0163-6804. doi: 10.1109/MCOM.2004.1341274. 19

Mustafa Y. El-Nainay. *Island Genetic Algorithm-based Cognitive Networks*. PhD thesis, Virginia Tech, July 2009. 36

Tamer ElBatt and Anthony Ephremides. Joint scheduling and power control for wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 3(1):74–85, January 2004. ISSN 1536-1276. doi: 10.1109/TWC.2003.819032. 24

Tamer A. ElBatt, Srikanth V. Krishnamurthy, Dennis Connors, and Son Dao. Power management for throughput enhancement in wireless ad-hoc networks. In Russell E. Trahan and Benjamin Leon, editors, *ICC 2000, IEEE International Conference on Communications*, pages 1506–1513. BellSouth, IEEE Communications Society, June 2000. ISBN 0-7803-6283-7. doi: 10.1109/ICC.2000.853748. 24, 25, 31

Thierry Ernst. *MobiWan: A ns-2.1b6 Simulation Platform for Mobile IPv6 in Wide Area Networks*. Motorola Labs Paris, May 2001. 79

Thierry Ernst. The information technology era of the vehicular industry. *SIGCOMM Computer Communication Review*, 36(2):49–52, 2006. ISSN 0146-4833. doi: 10.1145/1129582.1129595. 4

Thierry Ernst. IPv6 network mobility in the CVIS project. In *ITS 2007, 6th European Congress & Exhibition on Intelligent Transport Systems and Services*, June 2007. 12

Thierry Ernst and Arnaud de La Fortelle. Car-to-car and car-to-infrastructure communication system based on NEMO and MANET in IPv6. In *13th World Congress and Exhibition on Intelligent Transport Systems and Services*, October 2006. 4

Thierry Ernst and Hong-Yon Lach. Network mobility support terminology. RFC 4885, RFC Editor, July 2007. 23

ETSI EN 302 665. Intelligent transport systems (ITS); communications architecture. Version 1.1.1, September 2010. xvi, 4, 129

ETSI TS 102 636-3. Intelligent transport systems (ITS); vehicular communications; geonetworking; part 3: Network architecture, March 2010. 13

Euclid. Proposition I.47. In *Elements*, First Book. *c.* 300 BC.

Kevin Fall. A delay-tolerant network architecture for challenged internets. In Jon Crowcroft and David Wetherall, editors, *SIGCOMM 2003, Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM, August 2003. ISBN 1-58113-735-4. doi: 10.1145/863955.863960. xv, 2, 188

Dino Farinacci, Vince Fuller, Dave Meyer, and Darrel Lewis. Locator/ID separation protocol (LISP). Internet-Draft draft-ietf-lisp-13.txt, IETF Secretariat, June 2011a. 20

Dino Farinacci, Darrel Lewis, Dave Meyer, and Chris White. LISP mobile node. Technical Report draft-meyer-lisp-mn-05.txt, IETF Secretariat, May 2011b. 20

Marc E. Fiuczynski and Jeanna Matthews. PlanetLab: Overview, history, and future directions. *ACM SIGOPS Operating Systems Review*, 40:6–10, January 2006. ISSN 0163-5980. doi: 10.1145/1113361.1113366. 44

Sally Floyd. Metrics for the evaluation of congestion control mechanisms. RFC 5166, RFC Editor, March 2008. 14

Sally Floyd and Kevin Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, August 1999. ISSN 1063-6692. doi: 10.1109/90.793002. xxi, 14, 64, 75

Sally Floyd and Eddie Kohler. Profile for datagram congestion control protocol (DCCP) congestion control ID 2: TCP-like congestion control. RFC 4341, RFC Editor, March 2006. 15

Sally Floyd and Eddie Kohler. Profile for datagram congestion control protocol (DCCP) congestion ID 4: TCP-friendly rate control for small packets (TFRC-SP). RFC 5622, RFC Editor, August 2009. 15

Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. Equation-based congestion control for unicast applications. *SIGCOMM Computer Communication Review*, 30 (4):43–56, October 2000. ISSN 0146-4833. doi: 10.1145/347057.347397. xxi, 8, 15, 75

Sally Floyd, Eddie Kohler, and Jitendra Padhye. Profile for datagram congestion control protocol (DCCP) congestion control ID 3: TCP-friendly rate control (TFRC). RFC 4342, RFC Editor, March 2006. 15

Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. TCP friendly rate control (TFRC): Protocol specification. RFC 5348, RFC Editor, September 2008. xxi, xxii, 15, 75, 76, 77, 78, 79, 84, 85, 88, 95, 99, 190

Alan Ford, Costin Raiciu, and Mark Handley. TCP extensions for multipath operation with multiple addresses. Internet-Draft draft-ietf-mptcp-multiaddressed-02.txt, IETF Secretariat, July 2010. 19

Bryan Ford. Structured streams: A new transport abstraction. In Jun Murai and Kenjiro Cho, editors, *SIGCOMM 2007, Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 361–372. ACM, August 2007. ISBN 978-1-59593-713-1. doi: 10.1145/1282380.1282421. 30, 31

Bryan Ford and Janardhan Iyengar. Breaking up the transport logjam. In David Andersen and Steve Gribble, editors, *HotNets-VII, 7th ACM Workshop on Hot Topics in Networks*. ACM SIGCOMM, ACM, October 2008. 12

Bryan Ford and Janardhan Iyengar. Efficient cross-layer negotiation. In Will Leland and Ratul Mahajan, editors, *HotNets-VIII, 8th ACM Workshop on Hot Topics in Networks*. ACM SIGCOMM, ACM, October 2009. 30

Mat Ford, Mohammed Boucadair, Alain Durand, Pierre Levis, and Phil Roberts. Issues with IP address sharing. RFC 6269, RFC Editor, June 2011. 12

Mirko Franceschinis, Marco Mellia, Michela Meo, and Maurizio Munafò. Measuring TCP over WiFi: A real case. In Chadi Barakat, Elizabeth M. Belding-Royer, Andrew Campbell, Edward Knightly, Lakshman Krishnamurthy, Josep Mangues, Jitu Padhye, Konstantina Papagiannaki, Kave Salamatian, Aruna Seneviratne, Suresh Singh, and Leandros Tassiulas, editors, *WiNMee 2005, 1st workshop on Wireless Network Measurements*, April 2005. 18

Daniel H. Friend. *Cognitive Networks: Foundations to Applications*. PhD thesis, Virginia Tech, March 2009. 36

Daichi Funato, Kinuko Yasuda, and Hideyuki Tokuda. TCP-R: TCP mobility support for continuous operation. In Mostafa Ammar and Udaya Shankar, editors, *ICNP 1997, 5th International Conference on Network Protocols*, pages 229–236. Hitachi Telecommunications, IEEE Computer Society, October 1997. ISBN 0-8186-8061-X. doi: 10.1109/ICNP.1997.643720. 20

Baptiste Gaultier, Rayene Ben Rayana, and Jean-Marie Bonnin. Context management systems applied to mobility. In Marion Berbineau, Makoto Itami, and GuangJun Wen, editors, *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*, pages 330–335. IEEE Computer Society, October 2009. ISBN 1-4244-1178-5. 72

Vangelis Gazis, Nancy Alonistioti, and Lazaros Merakos. Toward a generic ”always best connected” capability in integrated WLAN/UMTS cellular mobile networks (and beyond). *IEEE Wireless Communications*, 12(3):20–29, June 2005. ISSN 1070-9916. doi: 10.1109/MWC.2005.1452851. 33, 35

GeoNet project. Final GeoNet architecture design. Deliverable GeoNet-D.1.2-v1.2, EC Information Society Technologies Programme, June 2010. 4, 13

Gene V. Glass, Percy D. Peckham, and James R. Sanders. Consequences of failure to meet assumptions underlying the fixed effects analyses of variance and covariance. *Review of Educational Research*, 42(3), 1972. ISSN 0034-6543. doi: 10.2307/1169991. 114

Tom Goff, James Moronski, Dhananjay S. Phatak, and Vipul Gupta. Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments. In Raphael Rom and Henning Schulzrinne, editors, *INFOCOM 2000, 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1537–1545. IEEE Computer Society, March 2000. ISBN 0-7803-5880-5. doi: 10.1109/INFCOM.2000.832552. 27, 76, 92, 96

Kazutaka Gogo, Rie Shibui, and Fumio Teraoka. An L3-driven fast handover mechanism in IPv6 mobility. In Vijay Kumar and Hiroyuki Morikawa, editors, *SAINT 2006,* *International Symposium on Applications on Internet, Workshops*, pages 10–13. IEEE Computer Society, January 2006. doi: 10.1109/SAINT-W.2006.6. 26

Fred Goldstein and John Day. Moving beyond TCP/IP. April 2010. 12

Maxim Graubner, Parag S. Mogre, Ralf Steinmetz, and Thorsten Lorenzen. A new QoE model and evaluation method for broadcast audio contribution over IP. In Dick Bulterman and Mohamed Hefeeda, editors, *NOSSDAV 2010, 20th international workshop on Network and operating systems support for digital audio and video*, pages 57–62. ACM, June 2010. ISBN 978-1-4503-0043-8. doi: 10.1145/1806565.1806581. 65

Luigi A. Grieco and Saverio Mascolo. Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control. *SIGCOMM Computer Communication Review*, 34(2):25–38, April 2004. ISSN 0146-4833. doi: 10.1145/997150.997155. 18

Ole Grøndalen, Pål Grønsund, Tor Breivik, and Paal Engelstad. Fixed WiMAX field trial measurements and analyses. In *16th IST Mobile and Wireless Communications Summit*, pages 1–5. IEEE Computer Society, July 2007. ISBN 1-4244-1662-0. doi: 10.1109/ISTMWC.2007.4299213. 16

Yan Grunenberger. *Réseaux sans fil de nouvelle génération : architectures spontanées et optimisations inter-couches*. PhD thesis, Institut Polytechnique de Grenoble, December 2008. 31, 32

Sri Gundavelli, Kent Leung, Vijay Devarapalli, Kuntal Chowdhury, and Basavaraj Patil. Proxy mobile IPv6. RFC 5213, RFC Editor, August 2008. 20

Qiang Guo, Jie Zhu, and Xianghua Xu. An adaptive multi-criteria vertical handoff decision algorithm for radio heterogeneous network. In Byeong G. Lee, editor, *ICC 2005, IEEE International Conference on Communications*, volume 4, pages 2769–2773. IEEE Communications Society, May 2005. ISBN 0-7803-8938-7. doi: 10.1109/ICC.2005.1494852. 33, 34, 35

Andrei Gurtov and Sally Floyd. Modeling wireless links for transport protocols. *SIGCOMM Computer Communication Review*, 34(2):85–96, April 2004. ISSN 0146-4833. doi: 10.1145/997150.997159. 96, 100

Eva Gustafsson and Annika Jonsson. Always best connected. *IEEE Wireless Communications*, 10(1):49–55, February 2003. ISSN 1536-1284. doi: 10.1109/MWC.2003.1182111. xv, xix, 2, 32, 47, 187

Eva Gustafsson, Annika Jonsson, and Charles E. Perkins. Mobile IPv4 regional registration. Internet-draft, RFC Editor, June 2004. 20

Youngjune Gwon, Daichi Funato, and Atsushi Takeshita. Adaptive approach for locally optimized IP handoffs across heterogeneous wireless networks. In Mario Gerla, editor, *MWCN 2002, 4th International Workshop on Mobile and Wireless Communications Network*, pages 475–479. IEEE Communications Society, Institute for Communications Research, IFIP TC-6, IEEE Communications Society, September 2002. ISBN 0-7803-7605-6. doi: 10.1109/MWCN.2002.1045810. 26, 33

Emir Halepovic, Qian Wu, Carey Williamson, and Majid Ghaderi. TCP over WiMAX: A measurement study. In Ethan Miller and Carey Williamson, editors, *MASCOTS 2008, 16th IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems*, pages 1–10. IEEE Computer Society, September 2008. ISBN 978-1-4244-2817-5. doi: 10.1109/MASCOT.2008.4770565. 16

Yunsop Han and Fumio Terakoa. SCTPmx: An SCTP fast handover mechanism using a single interface based on a cross-layer architecture. *IEICE Transactions on Communications*, E.92B(9):2864–2873, September 2009. ISSN 1745-1345. 27

Yunsop Han and Fumio Teraoka. SCTPfx: A fast failover mechanism based on cross-layer architecture in SCTP multihoming. In Kenjiro Cho, Martin May, and Jennifer Rexford, editors, *AINTEC 2008, 4th Asian Conference on Internet Engineering*, pages 113–122. ACM SIGCOMM, ACM, November 2008. ISBN 978-1-60558-127-9. doi: 10.1145/1503370.1503399. 27

Jérôme Härri, Fethi Filali, and Christian Bonnet. Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns. In Stefano Basagni, Antonio Capone, Luigi Fratta, and Giacomo Morabito, editors, *Med-Hoc-Net 2006, 5th Annual Mediterranean Ad Hoc Networking Workshop*. IFIP, June 2006. 4

David Harrington and Juergen Schoenwaelder. Transport subsystem for the simple network management protocol (SNMP). RFC 5590, RFC Editor, June 2009. 44

David Harrington, Randy Presuhn, and Bert Wijnen. An architecture for describing simple network management protocol (SNMP) management frameworks. RFC 3411, RFC Editor, December 2002. 44, 190

Michael R. Harwell, Elaine N. Rubinstein, William S. Hayes, and Corley C. Olds. Summarizing Monte Carlo results in methodological research: The one- and two-factor fixed effects ANOVA cases. *Journal of Educational and Behavioral Statistics*, 17(4):315–339, December 1992. doi: 10.3102/10769986017004315. 114

Stephen Hemminger. Network emulation with NetEm. In Martin Pool, editor, *LCA 2005, Australia's 6th national Linux conference (linux.conf.au)*. Linux Australia, Linux Australia, April 2005. 100, 109

Thomas R. Henderson. Host mobility for IP networks: A comparison. *IEEE Network*, 17(6):18–26, November 2003. ISSN 0890-8044. doi: 10.1109/MNET.2003.1248657. 21

Gavin Holland, Nitin H. Vaidya, and Paramvir Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In Mahmoud Naghshineh and Michele Zorzi, editors, *MobiCom 2001, 7th Annual International Conference on Mobile Computing and Networking*, pages 236–251. ACM SIGMOBILE, ACM, July 2001. ISBN 1-58113-422-3. doi: 10.1145/381677.381700. 23, 31

William S. Hortos. Cross-layer protocols optimized for real-time multimedia services in energy-constrained mobile ad hoc networks. In Raghuveer M. Rao, Soheil A. Dianat, and Michael D. Zoltowski, editors, *Digital Wireless Communications V, 5th Conference* *on Digital Wireless Communications*, volume 5100 of *Proceedings of SPIE*, pages 51–72. SPIE, April 2003. ISBN 0-8194-4960-1. doi: 10.1117/12.488493. 26, 32, 36

Mark Huang, Andy Bavier, and Larry Peterson. PlanetFlow: Maintaining accountability for network services. *ACM SIGOPS Operating Systems Review*, 40(1):89–94, January 2006. ISSN 0163-5980. doi: 10.1145/1113361.1113376. 44

Bert Hubert, Thomas Graf, Gregory Maxwell, Remco Van Mook, Martijn Van Oosterhout, Paul B. Schroeder, Jasper Spaans, and Pedro Larroy. *Linux Advanced Routing & Traffic Control HOWTO*. Linux Advanced Routing & Traffic Control, April 2004. 100

Geoff Huston. IPv4 address report. Daily report auto generated at 09-Dec-2011 07:59 UTC, December 2011. xvi, 2, 12

Norman C. Hutchinson and Larry L. Peterson. The x-Kernel: An architecture for implementing network protocols. *IEEE Transactions on Software Engineering*, 17(1):64–76, January 1991. ISSN 0098-5589. doi: 10.1109/32.67579. 30

Gianluca Iannaccone. CoMo: An open infrastructure for network monitoring — research agenda. Technical report, Intel Research, February 2005. 44

IEC 80000-13:2008. Quantities and units — part 13: Information science and technology. ISO/TC12 WG12, IEC/TC25, April 2008. 38, 44

IEEE Std 802.11-2007. IEEE standard for information technology — telecommunications and information exchange between systems — local and metropolitan area networks — specific requirements — part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, June 2007. 1, 15, 16, 26, 28, 52, 187, 190

IEEE Std 802.11p-2010. IEEE standard for information technology — telecommunications and information exchange between systems — local and metropolitan area networks — specific requirements — part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 6: Wireless access in vehicular environments, July 2010. 4

IEEE Std 802.16-2009. IEEE standard for local and metropolitan area networks — part 16: Air interface for broadband wireless access systems, May 2009. 1, 16, 17

IEEE Std 802.21-2008. IEEE standard for local and metropolitan area networks — part 21: Media independent handovers services, January 2009. xix, 30, 188

IEEE Std 802.3-2008. IEEE standard for information technology — telecommunications and information exchange between systems — local and metropolitan area networks — specific requirements — part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, 2008. 15, 16

IEEE Std 802.3ba-2010. IEEE standard for information technology — telecommunications and information exchange between systems — local and metropolitan area networks-specific requirements part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications amendment 4: Media access control parameters, physical layers and management parameters for 40 Gb/s and 100 Gb/s operation, June 2010. 16

Gianluca Insolvibile. The Linux socket filter: Sniffing bytes over the network. *Linux Journal*, 86, March 2001. ISSN 1075-3583. 43, 108

RTMaps SDK. *RTMaps Software Development Kit Version 3.0*. Intempora S.A., 2005. 44

ISO 21210:2011. Intelligent transport systems — communications access for land mobiles (CALM) — IPv6 networking, January 2011. xvi, 4

ISO 21217:2010. Intelligent transport systems — communications access for land mobiles (CALM) — architecture, April 2010. 30

ISO 25111:2009. Intelligent transport systems — communications access for land mobiles (CALM) — general requirements for using public networks, October 2009. 4, 187

ISO/CD 24102:2008. Intelligent transport systems — communications access for land mobiles (CALM) — CALM management, June 2009. xix, 30

ISO/IEC 7498-1:1994. Information technology — open systems interconnection — basic reference model: The basic model. Also published as ITU-T Recommendation X.200, November 1994. 1, 11, 189

ITU-T Recommendation G.1030. Estimating end-to-end performance in IP networks for data applications. ITU-T SG12, May 2006. 42

ITU-T Recommendation G.107. The E-model, a computational model for use in transmission planning. ITU-T SG12, March 2005. 41

ITU-T Recommendation G.1070. Opinion model for video-telephony applications. ITU-T SG12, April 2007. 41

ITU-T Recommendation P.10/G.100 Amendment 2. New definitions for inclusion in recommendation P.10/G.100. ITU-T SG12, July 2008. 40, 189

ITU-T Recommendation P.800. Methods for subjective determination of transmission quality. ITU-T SG12, August 1996. 40, 41, 189

Janardhan Iyengar and Bryan Ford. Flow splitting with fate sharing in a next generation transport services architecture. December 2009. 12

Philippe Jacquet, Paul Mühlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol for ad hoc networks. In Mohammad A. Iqbal and Mohammad A. Maud, editors, *INMIC 2001, IEEE International Multi Topic Conference*, pages 62–68. IEEE Computer Society, December 2002. ISBN 0-7803-7406-1. doi: 10.1109/INMIC.2001.995315. xv, 2

Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC-TR-301, Digital Equipment Corporation, September 1984. 63

Ravi Jain, James E. Burns, Michael Bereschinsky, and Charles Graff. Mobile IP with location registers (MIP-LR). Internet-Draft draft-jain-miplr-01.txt, IETF Secretariat, July 2001. 20

Gabor Jeney, Laszlo Bokor, and Zsigmond Mihaly. GPS aided predictive handover management for multihomed NEMO configurations. In Marion Berbineau, Makoto Itami, and GuangJun Wen, editors, *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*, pages 69–73. IEEE, October 2009. ISBN 978-1-4244-5346-7. doi: 10.1109/ITST.2009.5399380. 34

Mikael Johansson and Lin Xiao. Cross-layer optimization of wireless networks using nonlinear column generation. *IEEE Transactions on Wireless Communications*, 5(2): 435–445, February 2006. ISSN 1536-1276. doi: 10.1109/TWC.2006.1611067. 26, 36

David Johnson, Charles E. Perkins, and Jari Arkko. Mobility support in IPv6. RFC 3775, RFC Editor, June 2004. 79

David B. Johnson, Charles E. Perkins, and Jari Arkko. Mobility support in IPv6. RFC 6275, RFC Editor, July 2011. 20, 189

Guillaume Jourjon, Emmanuel Lochin, and Laurent Dairaine. Optimization of TFRC loss history initialization. *IEEE Communications Letters*, 11(3):276–278, March 2007. ISSN 1089-7798. doi: 10.1109/LCOMM.2007.061707. 77

Ad Kamerman and Leo Monteban. WaveLAN-II: A high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, 2(3):118–133, August 1997. ISSN 1089-7089. doi: 10.1002/bltj.2069. 23

Srikanth Kandula, Kate C. Lin, Tural Badirkhanli, and Dina Katabi. FatVAP: Aggregating AP backhaul capacity to maximize throughput. In Jon Crowcroft and Mike Dahlin, editors, *NSDI 2008, 5th USENIX Symposium on Networked Systems Design and Implementation*. USENIX, ACM SIGCOMM, ACM SIGOPS, USENIX Association, April 2008. 33, 35, 63

Thanasis Karapantelakis and Giorgos Iacovidis. Experimenting with real time applications in an IEEE 802.11b ad hoc network. In Hossam Hassanein and Marcel Waldvogel, editors, *LCN 2005, 30th IEEE Conference on Local Computer Networks*, volume 0, pages 554–559. IEEE Computer Society, November 2005. doi: 10.1109/LCN.2005.67. 16

Vikas Kawadia and P. R. Kumar. A cautionary perspective on cross-layer design. *IEEE Wireless Communications*, 12(1):3–11, February 2005. ISSN 1536-1284. doi: 10.1109/MWC.2005.1404568. xvii, xix, 6, 31, 48

Yacine Khaled, Manabu Tsukada, José Santa, Jinhyeock Choi, and Thierry Ernst. A usage oriented analysis of vehicular networks: from technologies to applications. *Journal of Communications*, 4(5):357–368, June 2009. ISSN 1796-2021. xvi, 3

Shoaib Khan, Yang Peng, Eckehard Steinbach, Marco Sgroi, and Worfgang Kellerer. Application-driven cross-layer optimization for video streaming over wireless networks. *IEEE Communications Magazine*, 44(1):122–130, January 2006. ISSN 0163-6804. doi: 10.1109/MCOM.2006.1580942. 28

Kalevi Kilkki. Quality of experience in communications ecosystem. *Journal of Universal Computer Science*, 14(5):615–624, March 2008. xx, 40, 47

Ingo Kofler, Joachim Seidl, Christian Timmerer, Hermann Hellwagner, Ismail Djama, and Toufik Ahmed. Using MPEG-21 for cross-layer multimedia content adaptation. *Signal, Image and Video Processing*, 2(4):355–370, December 2008. ISSN 1863-1703. doi: 10.1007/s11760-008-0088-x. 28

Eddie Kohler. Generalized connections in the datagram congestion control protocol. Internet-Draft draft-kohler-dccp-mobility-02.txt, IETF Secretariat, June 2006. 19, 20

Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, August 2000. ISSN 0734-2071. doi: 10.1145/354871.354874. 31

Eddie Kohler, Mark Handley, and Sally Floyd. Designing DCCP: Congestion control without reliability. *SIGCOMM Computer Communication Review*, 36(4):27–38, October 2006a. ISSN 0146-4833. doi: 10.1145/1151659.1159918. xxi, 8, 15, 75

Eddie Kohler, Mark Handley, and Sally Floyd. Datagram congestion control protocol (DCCP). RFC 4340, RFC Editor, March 2006b. xxi, 15, 39, 75, 94, 188

Eddie Kohler, Sally Floyd, and Arjuna Sathiaseelan. Faster restart for TCP friendly rate control (TFRC). Internet-Draft draft-ietf-dccp-tfrc-faster-restart-06.txt, IETF Secretariat, July 2008. 82, 94, 189

Samad S. Kolahi, Shaneel Narayan, Du Nguyen, and Yonathan Sunarto. Performance monitoring of various network traffic generators. In Richard Cant, editor, *UKSim 2011, 13th International Conference on Computer Modelling and Simulation*, pages 501–506. IEEE Computer Society, March 2011. ISBN 978-1-61284-705-4. doi: 10.1109/UKSIM.2011.102. 43

Rajeev Koodli. Mobile IPv6 fast handovers. RFC 5268, RFC Editor, June 2008. 22, 30

Rajeev Koodli and Charles E. Perkins. Fast handovers and context transfers in mobile networks. *SIGCOMM Computer Communication Review*, 31(5):37–47, October 2001. ISSN 0146-4833. doi: 10.1145/1037107.1037113. 90

Suresh Krishnan, Nicolas Montavont, Eric Njedjou, Siva Veerepalli, and Alper Yegin. Link-layer event notifications for detecting network attachments. RFC 4957, RFC Editor, August 2007. 26

Nandakishore Kushalnagar, Gabriel Montenegro, and Christian P. Schumacher. IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals. RFC 4919, RFC Editor, August 2007. 12

Arnaud de La Fortelle, Claude Laurgeau, Paul Muhlethaler, and Yasser Toor. Com2REACT: V2V communication for cooperative local traffic management, October 2007. 3

Sven Lahde and Lars Wolf. Dynamic network selection for robust communications: Why disruption tolerance matters. In Jörg Ott and Kun Tan, editors, *MobiArch 2009, 4th international workshop on Mobility in the evolving internet architecture*. ACM SIGMOBILE, ACM, June 2009. ISBN 978-1-60558-688-5/09/06. 33

Björn Landfeldt, Tomas Larsson, Yuri Ismailov, and Aruna Seneviratne. SLM, a framework for session layer mobility management. In Sudhir Dixit, Arun Somani, and Eun K. Park, editors, *ICCCN 1999, 8th International Conference on Computer Communications and Networks*, pages 452–456. Army Research Lab/Nokia, IEEE Communications Society, October 1999. ISBN 0-7803-5794-9. doi: 10.1109/ICCCN.1999.805557. 20

Conny Larsson, Michael Eriksson, Koshiro Mitsuya, Kazuyuki Tasaka, and Romain Kuntz. Flow distribution rule language for multi-access nodes. Internet-Draft draft-larsson-mext-flow-distribution-rules-02.txt, IETF Secretariat, February 2009. 22, 48

Lars-Åke Larzon, Mikael Degermark, Stephen Pink, Lars-Erik Jonsson, and Godred Fairhurst. The lightweight user datagram protocol (UDP-lite). RFC 3828, RFC Editor, July 2004. 13, 39

Cedric Launois and Marcelo Bagnulo. The paths toward IPv6 multihoming. *IEEE Communications Surveys & Tutorials*, 8(2):38–51, 2006. ISSN 1553-877X. doi: 10.1109/COMST.2006.315853. 21

Claude Laurgeau. *Le siècle de la voiture intelligente*. Mathématiques et informatique. Presse des Mines, November 2009. ISBN 978-2-911256-10-3. xvi, 3

Jean-Yves Le Boudec. *Performance Evaluation of Computer and Communication Systems*. EPFL Press, November 2010. ISBN 978-2-940222-40-7. 114

Eliot Lear and Ralph Droms. What's in a name: Thoughts from the NSRG. Internet-draft, IETF Secretariat, September 2003. 19

Heeyoung Lee, Seongkwan Kim, Okhwan Lee, Sunghyun Choi, and Sung J. Lee. Available bandwidth-based association in IEEE 802.11 wireless LANs. In Brahim Bensaou and Violet R. Syrotiuk, editors, *MSWiM 2008, 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 132–139. ACM SIGSIM, ACM, October 2008. ISBN 978-1-60558-235-1. doi: 10.1145/1454503.1454529. 33, 34

Jun S. Lee, Seok J. Koh, and Sang H. Kim. Analysis of handoff delay for mobile IPv6. In Tien M. Nguyen, editor, *VTC2004-Fall, 60th IEEE Vehicular Technology Conference*, volume 4, pages 2967–2969 Vol. 4. IEEE Computer Society, September 2004. ISBN 0-7803-8521-7. doi: 10.1109/VETECF.2004.1400604. 22, 90

Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. A brief history of the Internet. *SIGCOMM Computer Communication Review*, 39(5):22–31, October 2009. ISSN 0146-4833. doi: 10.1145/1629607.1629613. xv, 1

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22 (140):1–55, 1932. 73

Yu Lin, Shiduan Cheng, Wendong Wang, and Yuehui Jin. Measurement-based TFRC: Improving TFRC in heterogeneous mobile networks. *IEEE Transactions on Wireless Communications*, 5(8):1971–1975, August 2006. ISSN 1536-1276. doi: 10.1109/TWC.2006.1687706. 101

Sifeng Liu, Yi Lin, Xie Naiming, Jian Lirong, Dang Yaoguo, Fang Zhigeng, Su Chunhua, Zeng Bo, Wei Meng, and Yingjie Yang. Introduction to grey systems theory. In Sifeng Liu and Yi Lin, editors, *Grey Systems—Theory and Applications*, volume 68 of *Understanding Complex Systems*, chapter 1, pages 1–18. Springer-Verlag Berlin, 2011. ISBN 978-3-642-16157-5. doi: 10.1007/978-3-642-16158-2_1. 35

Xiaoshan Liu, Victor O. K. Li, and Ping Zhang. Joint radio resource management through vertical handoffs in 4G networks. In Zhi Ding and Chen-Nee Chuah, editors, *GlobeCom 2006, 49th IEEE Global Communications Conference*, pages 1–5. IEEE Communications Society, November 2006. ISBN 1-4244-0357-X. doi: 10.1109/GLOCOM.2006.338. 33, 34

Emmanuel Lochin, Laurent Dairaine, and Guillaume Jourjon. gTFRC, a TCP friendly QoS-aware rate control for DiffServ assured service. *Telecommunication Systems*, 33(1): 3–21, December 2006. ISSN 1018-4864. doi: 10.1007/s11235-006-9004-2. 27

Emmanuel Lochin, Guillaume Jourjon, Sébastien Ardon, and Patrick Senac. Promoting the use of reliable rate-based transport protocols: the Chameleon protocol. *International Journal of Internet Protocol Technology*, 5(4):175–189, 2010. ISSN 1743-8217. doi: 10.1504/IJIPT.2010.039229. 15, 92

Michael Loiacono, Jeffrey Johnson, Justinian Rosca, and Wade Trappe. Cross-layer link adaptation for wireless video. In Chengshan Xiao and Jan C. Olivier, editors, *ICC 2010, IEEE International Conference on Communications*, pages 1–6. IEEE Communications Society, May 2010. ISBN 978-1-4244-6402-9. doi: 10.1109/ICC.2010.5502650. 28

Yoann Lopez and Eric Robert. OpenMIH, an open-source media-independent handover implementation and its application to proactive pre-authentication. In Kostas Pentikousis, Oliver Blume, Ramón Agüero Calvo, and Symeon Papavassiliou, editors, *Mobile Networks and Management*, volume 32 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, chapter 2, pages 14–25–25. Springer-Verlag Berlin, 2010. ISBN 978-3-642-11816-6. doi: 10.1007/978-3-642-11817-3_2. 30

Jean Lorchat and Keisuke Uehara. Optimized inter-vehicle communications using NEMO and MANET. In Onur Altintas and Wai Chen, editors, *MobiQuitous 2006, 3rd Annual International Conference on Mobile and Ubiquitous Systems, Workshops*, volume 0, pages 1–6. IEEE Computer Society, June 2006. ISBN 0-7803-9791-6. doi: 10.1109/MOBIQW.2006.361762. 4

Juha-Pekka Mäkelä, Mika Ylianttila, and Kaveh Pahlavan. Handoff decision in multi-service networks. In Hamid Aghvami, editor, *PIMRC 2000, 11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, volume 2, pages 655–659. IEEE Communications Society, September 2000. ISBN 0-7803-6463-5. doi: 10.1109/PIMRC.2000.881503. 26, 33

David A. Maltz and Pravin Bhagwat. MSOCKS: An architecture for transport layer mobility. In Ian F. Akyildiz, editor, *INFOCOM 1998, 17th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1037–1045. IEEE, March 1998. ISBN 0-7803-4383-2. doi: 10.1109/INFCOM.1998.662913. 20

Jukka Manner and Markku Kojo. Mobility related terminology. RFC 3753, RFC Editor, June 2004. 23

Kim Marriott, Nicholas Nethercote, Reza Rafeh, Peter J. Stuckey, María García de la Banda, and Mark Wallace. The design of the Zinc modelling language. *Constraints*, 13 (3):229–267, September 2008. ISSN 1383-7133. doi: 10.1007/s10601-008-9041-4. 65

Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Yahya Sanadidi, and Ren Wang. TCP Westwood: Bandwidth estimation for enhanced transport over wireless links. In Mahmoud Naghshineh and David B. Johnson, editors, *MobiCom 2001, 7th annual international conference on Mobile Computing and networking*, pages 287–297. ACM SIGMOBILE, ACM, July 2001. ISBN 1-58113-422-3. doi: 10.1145/381677.381704. 18

Matt Mathis. Pushing up performance for everyone. Presentation to Joint Techs, December 1999. 5

Matt Mathis, John Heffner, and Raghu Reddy. Web100: Extended TCP instrumentation for research, education and diagnosis. *SIGCOMM Computer Communication Review*, 33 (3):69–79, July 2003. ISSN 0146-4833. doi: 10.1145/956993.957002. xxiii, 29, 43, 103

Matt Mathis, John Heffner, and Rajiv Raghunarayan. TCP extended statistics MIB. RFC 4898, RFC Editor, May 2007. 44

Arifumi Matsumoto, Masahiro Kozuka, Kenji Fujikawa, and Yasuo Okabe. TCP multihome options. Internet-Draft draft-arifumi-tcp-mh-00.txt, IETF Secretariat, October 2003. 20

Nils-Erik Mattsson. A DCCP module for ns-2. Master's thesis, Luleå Tekniska Universitet, May 2004. 79, 95

Pete McCann. Mobile IPv6 fast handovers for 802.11 networks. RFC 4260, RFC Editor, November 2005. 22

Steven McCanne and Van Jacobson. The BSD packet filter: A new architecture for user-level packet capture. In *USENIX Winter 1993*, page 2. USENIX Association, 1993. 43

Olivier Mehani and Roksana Boreli. Adapting TFRC to mobile networks with frequent disconnections. In Keith W. Ross and Leandros Tassiulas, editors, *CoNEXT 2008, 4th ACM International Conference on emerging Networking EXperiments and Technologies, Student Workshop*. ACM SIGCOMM, ACM, December 2008. ISBN 978-1-60558-210-8. doi: 10.1145/1544012.1544049.

Olivier Mehani, Rodrigo Benenson, Séverin Lemaignan, and Thierry Ernst. Networking needs and solutions at Imara. In Ulrich Finger, Masayuki Fujise, Christian Bonnet, Massimiliano Lenardi, Shozo Komaki, and Guangjun Wen, editors, *ITST 2007, 7th*

International Conference on Intelligent Transport Systems Telecommunications, pages 362–367. IEEE Computer Society, 2007. doi: 10.1109/ITST.2007.4295894. 4

Olivier Mehani, Roksana Boreli, and Thierry Ernst. Context-adaptive vehicular network optimization. In Marion Berbineau, Makoto Itami, and GuangJun Wen, editors, *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*, pages 186–191. IEEE Computer Society, October 2009a. ISBN 1-4244-1178-5.

Olivier Mehani, Roksana Boreli, and Thierry Ernst. Analysis of TFRC in disconnected scenarios and performance improvements with Freeze-DCCP. In Jörg Ott and Kun Tan, editors, *MobiArch 2009, 4th International Workshop on Mobility in the Evolving Internet Architecture*. ACM SIGMOBILE, ACM, June 2009b. ISBN 978-1-60558-688-5/09/06.

Olivier Mehani, Roksana Boreli, Guillaume Jourjon, and Thierry Ernst. Mobile multimedia streaming improvements with Freeze-DCCP. In Romit R. Choudhury and Henrik Lundgren, editors, *MobiCom 2010, 16th Annual International Conference on Mobile Computing and Networking, Demonstration Session*. ACM SIGMOBILE, September 2010.

Olivier Mehani, Roksana Boreli, Michael Maher, and Thierry Ernst. User- and application-centric multihomed flow management. In Tom Pfeifer and Anura Jayasumana, editors, *LCN 2011, 36th IEEE Conference on Local Computer Networks*, pages 26–34. IEEE Computer Society, IEEE Computer Society, October 2011a.

Olivier Mehani, Guillaume Jourjon, Jolyon White, Thierry Rakotoarivelo, Roksana Boreli, and Thierry Ernst. Characterisation of the effect of a measurement library on the performance of instrumented tools. Technical Report 4879, NICTA, May 2011b.

Archan Misra, Subir Das, Anthony McAuley, Ashutosh Dutta, and Sajal K. Das. IDMP: An intra-domain mobility management protocol using mobility agents. Internet-Draft draft-misra-mobileip-idmp-00.txt, IETF Secretariat, July 2000. 20

Archan Misra, Subir Das, Ashutish Dutta, Anthony McAuley, and Sajal K. Das. IDMP-based fast handoffs and paging in IP-based 4G mobile networks. *IEEE Communications Magazine*, 40(3):138–145, March 2002. ISSN 0163-6804. doi: 10.1109/35.989774. 20

Joseph Mitola. *Cognitive Radio — An Integrated Agent Architecture for Software Defined Radio*. DTech thesis, Royal Institute of Technology (KTH), May 2000. 31

Joseph Mitola and Gerald Q. Maguire. Cognitive radio: Making software radios more personal. *IEEE Personal Communications*, 6(4):13–18, August 1999. ISSN 1070-9916. doi: 10.1109/98.788210. 31

Koshiro Mitsuya, Romain Kuntz, Shinta Sugimoto, Ryuji Wakikawa, and Jun Murai. A policy management framework for flow distribution on multihomed end nodes. In Xiaoming Fu, Katherine Guo, Sue Moon, and Ryuji Wakikawa, editors, *MobiArch 2007, ACM/IEEE international workshop on Mobility in the evolving Internet Architecture*. ACM SIGMOBILE, ACM, August 2007. ISBN 978-1-59593-784-8. doi: 10.1145/1366919.1366933. 23, 48

MobileMAN project. MobileMAN architecture, protocols and services. Deliverable MobileMAN-D.10, EC Information Society Technologies Programme, October 2004. 29

Shantidev Mohanty and Ian F. Akyildiz. A cross-layer (layer 2 + 3) handoff management protocol for next-generation wireless systems. *IEEE Transactions on Mobile Computing*, 5(10):1347–1360, October 2006. ISSN 1536-1233. doi: 10.1109/TMC.2006.142. 26, 33

Nicolas Montavont and Thomas Noël. Stronger interaction between link layer and network layer for an optimized mobility management in heterogeneous IPv6 networks. *Pervasive and Mobile Computing*, 2(3):233–261, September 2006. ISSN 1574-1192. doi: 10.1016/j.pmcj.2006.02.001. 26, 76

Tim Moors. Protocol organs: Modularity should reflect function, not timing. pages 91–100. IEEE Communications Society, April 1998. ISBN 0-7803-4783-8. doi: 10.1109/OPNARC.1998.662046. 30

Stephen Morley and Malcolm Adams. Some simple statistical tests for exploring single-case time-series data. *British Journal of Clinical Psychology*, 28(1):1–18, February 1989. ISSN 0144-6657. 114

Robert Moskowitz, Pekka Nikander, Petri Jokela, and Tom Henderson. Host identity protocol. RFC 5201, RFC Editor, April 2008. 21, 188

Jayanth Mysore and Vaduvur Bharghavan. A new multicasting-based architecture for internet host mobility. In David B. Johnson and Christopher Rose, editors, *MobiCom 1997, 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 161–172. ACM, September 1997. ISBN 0-89791-988-2. doi: 10.1145/262116.262144. 20

Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In Marco Conti and Raghupathy Sivakumar, editors, *MobiHoc 2006, 7th ACM international symposium on Mobile Ad Hoc Networking and Computing*, pages 108–119. ACM SIGMOBILE, ACM, May 2006. ISBN 1-59593-368-9. doi: 10.1145/1132905.1132918. 4

Fawad Nazir and Aruna Seneviratne. Towards mobility enabled protocol stack for future wireless network. *Ubiquitous Computing and Communication Journal*, 2(4), August 2007. xix, 19

Anthony J. Nicholson, Yatin Chawathe, Mike Y. Chen, Brian D. Noble, and David Wetherall. Improved access point selection. In Mahadev Satyanarayanan and Nigel Davies, editors, *MobiSys 2006, 4th international conference on Mobile systems, applications and services*, MobiSys '06, pages 233–245. ACM SIGMOBILE, ACM, June 2006. ISBN 1-59593-195-3. doi: 10.1145/1134680.1134705. 33

Anthony J. Nicholson, Scott Wolchok, and Brian D. Noble. Juggler: Virtual networks for fun and profit. *IEEE Transactions on Mobile Computing*, 9(1):31–43, January 2010. ISSN 1536-1233. doi: 10.1109/TMC.2009.97. 35

Erik Nordmark and Marcelo Bagnulo. Multihoming L3 shim approach. Internet-Draft draft-ietf-multi6-l3shim-00.txt, IETF Secretariat, January 2005. 21

Tanır Özçelebi, M. Oğuz Sunay, Tekalp, and M. Reha Civanlar. Cross-layer optimized rate adaptation and scheduling for multiple-user wireless video streaming. *IEEE Journal on Selected Areas in Communications*, 25(4):760–769, May 2007. ISSN 0733-8716. doi: 10.1109/JSAC.2007.070512. 29

Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling TCP throughput: A simple model and its empirical validation. *SIGCOMM Computer Communication Review*, 28(4):303–314, October 1998. ISSN 0146-4833. doi: 10.1145/285243.285291. xxi, 15, 54, 77

Jeffrey Pang, Ben Greenstein, Michael Kaminsky, Damon McCoy, and Srinivasan Seshan. Wifi-Reports: Improving wireless network selection with collaboration. In Jason Flinn and Anthony LaMarca, editors, *MobiSys 2009, 7th international conference on Mobile Systems, applications, and services*, pages 123–136. ACM SIGMOBILE, ACM, June 2009. ISBN 978-1-60558-566-6. doi: 10.1145/1555816.1555830. 33, 34

Hyooon Park, Sunghoon Yoon, Taehyoun Kim, Jungshin Park, Misun Do, and Jaiyong Lee. Vertical handoff procedure and algorithm between IEEE802.11 WLAN and CDMA cellular network. In Jaiyong Lee and Chul-Hee Kang, editors, *Mobile Communications*, volume 2524 of *Lecture Notes in Computer Science*, chapter 11, pages 103–112. Springer-Verlag Berlin, March 2003. ISBN 978-3-540-00732-6. doi: 10.1007/3-540-36555-9_11. 26, 33

KyoungSoo Park and Vivek S. Pai. CoMon: A mostly-scalable monitoring system for PlanetLab. *ACM SIGOPS Operating Systems Review*, 40:65–74, January 2006. ISSN 0163-5980. doi: 10.1145/1113361.1113374. 44

Minu Park, Jaehyung Lee, Jahwan Koo, and Hyunseung Choo. Freeze TCPv2: An enhancement of Freeze TCP for efficient handoff in heterogeneous networks. In Gavriel Salvendy and Michael Smith, editors, *Human Interface and the Management of Information. Information and Interaction*, volume 5618 of *Lecture Notes in Computer Science*, chapter 49, pages 448–457. Springer-Verlag Berlin, 2009. ISBN 978-3-642-02558-7. doi: 10.1007/978-3-642-02559-4_49. 27

Vern Paxson, Guy Almes, Jamshid Mahdavi, and Matt Mathis. Framework for IP performance metrics. RFC 2330, RFC Editor, May 1998. xx, 37

Eranga Perera, Vijay Sivaraman, and Aruna Seneviratne. Survey on network mobility support. *SIGMOBILE Mobile Computing and Communications Review*, 8(2):7–19, April 2004. ISSN 1559-1662. doi: 10.1145/997122.997127. 20

Charles E. Perkins. IP mobility support for IPv4. RFC 3344, RFC Editor, August 2002. 20, 189

Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In Ramon Caceres, editor, *WMCSA 1999, 2nd IEEE Workshop on Mobile Computing*

Systems and Applications, pages 90–100. IEEE Computer Society, 1999. ISBN 0-7695-0025-0. doi: 10.1109/MCSA.1999.749281. xv, 2

Karine Perset. Internet addressing: Measuring deployment of IPv6. Technical Report DSTI/ICCP/CISP(2009)17/FINAL, OECD, April 2010. 12

Henrik Petander. Energy-aware network selection using traffic estimation. In Songwu Lu and Hewu Li, editors, *MICNET 2009, 1st ACM workshop on Mobile Internet through Cellular Networks*, pages 55–60. ACM SIGMOBILE, ACM, September 2009. ISBN 978-1-60558-753-0. doi: 10.1145/1614255.1614268. xxii, 33, 34, 47, 57, 58, 66, 103

Kandaraj Piamrat, César Viho, Adlen Ksentini, and Jean-Marie Bonnin. QoE-aware network selection in wireless heterogeneous networks. Technical Report RR-7282, Inria, May 2010. 33, 34, 48

Saar Pilosof, Ramachandran Ramjee, Danny Raz, Yuval Shavitt, and Prasun Sinha. Understanding TCP fairness over wireless LAN. In Jim Roberts and Ness Shroff, editors, *INFOCOM 2003, 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 863–872. IEEE Computer Society, April 2003. doi: 10.1109/INFCOM.2003.1208924. 18

Esa Piri and Kostas Pentikousis. IEEE 802.21. *The Internet Protocol Journal*, 12(2): 7–27, June 2009a. 30

Esa Piri and Kostas Pentikousis. Towards a GNU/Linux IEEE 802.21 implementation. In Gerhard Fettweis, Wojciech Kabacinski, and Heinrich Stüttgen, editors, *ICC 2009, IEEE International Conference on Communications*, pages 1–5. IEEE Communications Society, June 2009b. doi: 10.1109/ICC.2009.5199534. 30

Jonathan B. Postel. User datagram protocol. RFC 768, RFC Editor, August 1980. 13, 190

Jonathan B. Postel. Transmission control protocol. RFC 793, RFC Editor, September 1981. 14, 190

Randy Presuhn, Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose, and Steven Waldbusser. Management information base (MIB) for the simple network management protocol (SNMP). RFC 3418, RFC Editor, December 2002. 44

Jarmo Prokkola, Pekka H. J. Perälä, Mikko Hanski, and Esa Piri. 3G/HSPA performance in live networks from the end user perspective. In Gerhard Fettweis, editor, *ICC 2009, IEEE International Conference on Communications*, pages 1–6. TU Dresden, IEEE, June 2009. ISBN 978-1-4244-3435-0. doi: 10.1109/ICC.2009.5198575. 16, 55

Ioannis Psaras and Lefteris Mamatas. On demand connectivity sharing: Queuing management and load balancing for user-provided networks. *Computer Networks*, 55(2): 399–414, February 2011. ISSN 1389-1286. doi: 10.1016/j.comnet.2010.08.015. 34

Jani Puttonen, Gabor Fekete, Jukka Makela, Timo Hamalainen, and Jorma Narikka. Using link layer information for improving vertical handovers. In Bernhard Walke, Klaus

David, Martin Haardt, and Petri Mahonen, editors, *PIMRC 2005, 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 3, pages 1747–1752. IEEE Computer Society, September 2005. ISBN 9783800729098. doi: 10.1109/PIMRC.2005.1651742. 27

Liang Qin and Thomas Kunz. Survey on mobile ad hoc network routing protocols and cross-layer design. Technical Report SCE-04-14, Carleton University, August 2004. 25

Ahmad Rahmati and Lin Zhong. Context-for-wireless: Context-sensitive energy-efficient wireless data transfer. In Gaetano Borriello and Ramón Cáceres, editors, *MobiSys 2007, 5th international conference on Mobile systems, applications and services*, pages 165–178. ACM SIGMOBILE, USENIX, ACM, June 2007. ISBN 978-1-59593-614-1. doi: 10.1145/1247660.1247681. 33, 34

Thierry Rakotoarivelo, Maximilian Ott, Guillaume Jourjon, and Ivan Seskar. OMF: A control and management framework for networking nestbeds. *SIGOPS Operating Systems Review*, 43(4):54–59, January 2010. ISSN 0163-5980. doi: 10.1145/1713254.1713267. 98, 105, 189

Wassim Ramadan, Eugen Dedu, and Julien Bourgeois. EcnLD, ECN loss differentiation to optimize the performance of transport protocols on wireless networks. In Alexey Vinel and Qiang Ni, editors, *ICUMT 2009, International Conference on Ultra Modern Telecommunications & Workshops*, pages 1–6. IEEE Communications Society, October 2009. doi: 10.1109/ICUMT.2009.5345369. 27

Kadangode K. Ramakrishnan, Sally Floyd, and David L. Black. The addition of explicit congestion notification (ECN) to IP. RFC 3168, RFC Editor, September 2001. 27, 188

Ramachandran Ramjee, Thomas F. La Porta, Sandra R. Thuel, Kannan Varadhan, and Luca Salgarelli. IP micro-mobility support using HAWAII. Internet-Draft draft-ietf-mobileip-hawaii-01.txt, IETF Secretariat, July 2000. 20

Ramachandran Ramjee, Kannan Varadhan, Luca Salgarelli, Sandra R. Thuel, Shie-Yuan Wang, and Thomas La Porta. HAWAII: A domain-based approach for supporting mobility in wide-area wireless networks. *IEEE/ACM Transactions on Networking*, 10(3): 396–410, June 2002. ISSN 1063-6692. doi: 10.1109/TNET.2002.1012370. 20

Upendra Rathnayake and Max Ott. Predicting network availability using user context. In Liviu Iftode, editor, *Mobiquitous 2008, 5th Annual International Conference on Mobile and Ubiquitous Systems*, pages 1–8. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008. ISBN 978-963-9799-27-1. doi: 10.4108/ICST.MOBIQUITOUS2008.3563. xxii, 34, 103

Pierre Reinbold and Olivier Bonaventure. Ip micro-mobility protocols. *IEEE Communications Surveys & Tutorials*, 5(1):40–57, 2003. ISSN 1553-877X. doi: 10.1109/COMST.2003.5342229. 20

Maximilian Riegel and Michael Tuexen. Mobile sctp. Internet-Draft draft-riegel-tuexen-mobile-sctp-09.txt, IETF Secretariat, November 2007. 20

Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve Schooler. SIP: Session initiation protocol. RFC 3261, RFC Editor, June 2002. 20, 190

Thomas L. Saaty. *Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process*, volume 6 of *The Analytic Hierarchy Process*. RWS Publications, 2000. ISBN 0-9620317-6-3. 34, 73

Bahareh Sadeghi, Vikram Kanodia, Ashutosh Sabharwal, and Edward Knightly. OAR: An opportunistic auto-rate media access protocol for ad hoc networks. *Wireless Networks*, 11(1-2):39–53, January 2005. ISSN 1022-0038. doi: 10.1007/s11276-004-4745-x. 24

Werayut Saesue, Chun T. Chou, and Jian Zhang. Cross-layer QoS-optimized EDCA adaptation for wireless video streaming. In Wai-Kuen Cham and Fernando Pereira, editors, *ICIP 2010, 17th IEEE International Conference on Image Processing*, pages 2925–2928. IEEE Computer Signal Processing Society, September 2010. ISBN 978-1-4244-7994-8. doi: 10.1109/ICIP.2010.5652233. 28

Jerry H. Saltzer, David P. Reed, and David D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288, November 1984. ISSN 0734-2071. doi: 10.1145/357401.357402. 12

José Santa, Manabu Tsukada, Thierry Ernst, Olivier Mehani, and Antonio F. Gómez-Skarmeta. Assessment of VANET multi-hop routing over an experimental platform. *International Journal of Internet Protocol Technology*, 4(3):158–172, September 2009. ISSN 1743-8209. doi: 10.1504/IJIPT.2009.028655.

Pasi Sarolahti, Sally Floyd, and Markku Kojo. Transport-layer considerations for explicit cross-layer indications. Internet-Draft draft-sarolahti-tsvwg-crosslayer-01.txt, IETF Secretariat, March 2007. 27

Golam Sarwar, Roksana Boreli, and Emmanuel Lochin. Xstream-x264: A tool for real-time H.264 encoding and streaming with cross-layer integration. In Jin Li, Philippe Salembier, Dinei Florencio, Mohamed Hefeeda, Alex Loui, and Sethuraman Panchanathan, editors, *ICME 2011, 12th IEEE International Conference on Multimedia & Expo*. IEEE Communications Society, IEEE Computer Society, IEEE Signal Processing Society, IEEE, July 2011. 28

Henning Schulzrinne and Elin Wedlund. Application-layer mobility using SIP. *SIGMOBILE Mobile Computing and Communications Review*, 4(3):47–57, July 2000. ISSN 1559-1662. doi: 10.1145/372346.372369. 20

Henning Schulzrinne, Stephen L. Casner, Ron Frederick, and Van Jacobson. RTP: A transport protocol for real-time applications. RFC 1889, RFC Editor, January 1996. 39

SeVeCom project. Security architecture and mechanisms for V2V/V2I. Deliverable SeVeCom-D.2.1-v3.0, EC Information Society Technologies Programme, February 2008. 3

Sanjay Shakkottai, Theodore S. Rappaport, and Peter C. Karlsson. Cross-layer design for wireless networks. *IEEE Communications Magazine*, 41(10):74–80, October 2003. ISSN 0163-6804. doi: 10.1109/MCOM.2003.1235598. 5, 23

Jatinder Singh, Tansu Alpcan, Piyush Agrawal, and Varun Sharma. A Markov decision process based flow assignment framework for heterogeneous network access. *Wireless Networks*, 16(2):481–495, February 2010. ISSN 1022-0038. doi: 10.1007/s11276-008-0148-8. 35

Alex C. Snoeren and Hari Balakrishnan. TCP connection migration. Internet-Draft draft-snoeren-tcp-migrate-00.txt, IETF Secretariat, November 2000. 20

Rute Sofia and Paulo Mendes. User-provided networks: Consumer as provider. *IEEE Communications Magazine*, 46(12):86–91, December 2008. ISSN 0163-6804. doi: 10.1109/MCOM.2008.4689212. xv, 2, 34, 190

Hesham Soliman, Claude Castelluccia, Karim El Malki, and Ludovic Bellier. Hierarchical mobile IPv6 mobility management (HMIPv6). RFC 4140, RFC Editor, August 2005. 20

Qingyang Song and Abbas Jamalipour. Network selection in an integrated wireless LAN and UMTS environment using mathematical modeling and computing techniques. *IEEE Wireless Communications*, 12(3):42–48, June 2005. ISSN 1070-9916. doi: 10.1109/MWC.2005.1452853. 33, 34, 35, 47

Mahesh Sooriyabandara, Tim Farnham, Matthias Wellens, Janne Riihijärvi, Petri Mähönen, Alain Gefflaut, José A. Galache, Diego Melpignano, and Arthur van Rooijen. Unified link layer API: A generic and open API to manage wireless media access. *Computer Communications*, 31(5):962–979, March 2008. ISSN 0140-3664. doi: 10.1016/j.comcom.2007.12.025. 27

Robert J. Sparks. The session initiation protocol (SIP) refer method. RFC 3515, RFC Editor, April 2003. 20, 190

Vineet Srivastava and Mehul Motani. Cross-layer design: A survey and the road ahead. *IEEE Communications Magazine*, 43(12):112–119, December 2005. ISSN 0163-6804. doi: 10.1109/MCOM.2005.1561928. xvii, xix, 5, 23, 24, 25, 28, 29, 30, 183

Rafal Stankiewicz, Piotr Cholda, and Andrzej Jajszczyk. QoX: What is it really? *IEEE Communications Magazine*, 49(4):148–158, April 2011. ISSN 0163-6804. doi: 10.1109/MCOM.2011.5741159. xx, 37, 40, 183

Randall R. Stewart. Stream control transmission protocol. RFC 4960, RFC Editor, September 2007. 14, 190

Randall R. Stewart, Michael A. Ramalho, Qiaobing Xie, Michael Tuexen, and Phillip T. Conrad. Stream control transmission protocol (SCTP) partial reliability extension. RFC 3758, RFC Editor, May 2004. 15

Michael Still. *The Definitive Guide to ImageMagick*. Apress, December 2005. ISBN 978-1590595904. 44

Lucian Suciu, Jean-Marie Bonnin, Karine Guillouard, and Thierry Ernst. Multiple network interfaces management for mobile routers. In André Chomette and Shozo Komaki, editors, *ITST 2005, 5th International Conference on ITS Telecommunications*. Institut Télécom—Télécom Bretagne, Association Technopôle Brest—Iroise, June 2005. 35

Yi Sun, Yuming Ge, Jue Yuan, Jihua Zhou, Stephen Herborn, and Dongdong Chen. Pawes: A flow distribution algorithm based on priority and weight self-production. In Moe Z. Win, Hamid Aghvami, Andrea Conti, and Henk Wymeersch, editors, *WCNC 2009, IEEE Wireless Communications and Networking Conference*, pages 1–6. IEEE Communications Society, April 2009. ISBN 978-1-4244-2947-9. doi: 10.1109/WCNC.2009.4917884. 33, 35

Fumio Teraoka, Masahiro Ishiyama, and Mitsunobu Kunishi. LIN6: A solution to mobility and multi-homing in IPv6. Internet-Draft draft-teraoka-ipng-lin6-02.txt, IETF Secretariat, June 2003. 20

Fumio Teraoka, Kazutaka Gogo, Koshiro Mitsuya, Rie Shibui, and Koki Mitani. Unified layer 2 (L2) abstractions for layer 3 (L3)-driven fast handover. RFC 5184, RFC Editor, May 2008. xix, 26, 27, 30

Ryan W. Thomas. *Cognitive Networks*. PhD thesis, Virginia Tech, June 2007. 6, 36

Ryan W. Thomas, Luiz A. DaSilva, and Allen B. MacKenzie. Cognitive networks. In Dale Hatfield and Preston Marshall, editors, *DySPAN 2005, 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 352–360. IEEE Computer Society, November 2005. ISBN 1-4244-0013-9. doi: 10.1109/DYSPAN.2005.1542652. 31

Ryan W. Thomas, Daniel H. Friend, Luiz A. DaSilva, and Allen B. Mackenzie. Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine*, 44(12):51–57, December 2006. ISSN 0163-6804. doi: 10.1109/MCOM.2006.273099. 31

Nathanael Thompson, Guanghui He, and Haiyun Luo. Flow scheduling for end-host multihoming. In Arturo Azcorra, Joe Touch, and Zhili Zhang, editors, *INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, pages 1–12. IEEE, April 2006. ISBN 1-4244-0221-2. doi: 10.1109/INFOCOM.2006.197. 35

Hongtao Tian, S. K. Bose, Choi L. Law, and Wendong Xiao. CLA-QOS: A cross-layer QoS provisioning approach for mobile ad-hoc networks. In Hugh Bradlow, Richard Harris, and Rao Kotagiri, editors, *Tencon 2005, IEEE Conference on Convergent Technologies for Asia-Pacific Region*, pages 1–6. IEEE Computer Society, November 2005. ISBN 0-7803-9311-2. doi: 10.1109/TENCON.2005.300897. 26

Brian Tierney, Dan Gunter, Jason Lee, Martin Stoufer, and Joseph B. Evans. Enabling network-aware applications. In William Johnston, editor, *HPDC-10 (2001), 10th IEEE International Symposium on High Performance Distributed Computing*, pages 281–288. IEEE Computer Society Technical Committee on Distributed Processing, IEEE Computer Society, April 2001. ISBN 0-7695-1296-8. doi: 10.1109/HPDC.2001.945196. 29

Ajay Tirumala, Les Cottrell, and Tom Dunigan. Measuring end-to-end bandwidth with Iperf using Web100. In *PAM 2003, Passive and Active Monitoring Workshop*, number SLAC-PUB-9733, April 2003. 43

Jean Tourrilhes. L7-mobility: A framework for handling mobility at the application level. In Ramon Agustí and Oriol Sallent, editors, *PIMRC 2004, 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1246–1251. Universitat Politècnica de Catalunya, IEEE Communications Society, September 2004. ISBN 0-7803-8523-3. doi: 10.1109/PIMRC.2004.1373897. 20

George Tsirtsis, Hesham Soliman, Nicolas Montavont, Gerardo Giaretta, and Koojana Kuladinithi. Flow bindings in mobile IPv6 and network mobility (NEMO) basic support. RFC 6089, RFC Editor, January 2011. 22, 48

Manabu Tsukada, Olivier Mehani, and Thierry Ernst. Simultaneous usage of NEMO and MANET for vehicular communication. In Miguel P. de Leon, editor, *TridentCom 2008, 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), March 2008. ISBN 978-963-9799-24-0. 35

Manabu Tsukada, José Santa, Olivier Mehani, Yacine Khaled, and Thierry Ernst. Design and experimental evaluation of a vehicular network based on NEMO and MANET. *EURASIP Journal on Advances in Signal Processing*, 2010:1–18, September 2010. doi: 10.1155/2010/656407. 44

András G. Valkó. Cellular IP: A new approach to Internet host mobility. *SIGCOMM Computer Communication Review*, 29(1):50–65, January 1999. ISSN 0146-4833. doi: 10.1145/505754.505758. 20

Mihaela van der Schaar, Santhana Krishnamachari, Sunghyun Choi, and Xiaofeng Xu. Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 wlans. *IEEE Journal on Selected Areas in Communications*, 21(10):1752–1763, December 2003. ISSN 0733-8716. doi: 10.1109/JSAC.2003.815231. 28

Pablo Vidales, Javier Baliosian, Joan Serrat, Glenford Mapp, Frank Stajano, and Andy Hopper. Autonomic system for mobility support in 4G networks. *IEEE Journal on Selected Areas in Communications*, 23(12):2288–2304, December 2005. ISSN 0733-8716. doi: 10.1109/JSAC.2005.857198. 34

ns-2 manual. *The ns Manual (formerly ns Notes and Documentation)*. VINT Project, January 2009. 78

Ryuji Wakikawa, Kouji Okada, Rajeev Koodli, and Anders Nilsson. Design of vehicle network: mobile gateway for MANET and NEMO converged communication. In David B. Johnson and Raja Sengupta, editors, *VANET 2005, 2nd ACM international workshop on Vehicular Ad hoc Networks*, pages 81–82. ACM SIGMOBILE, ACM, 2005. ISBN 1-59593-141-4. doi: 10.1145/1080754.1080768. 4

Ryuji Wakikawa, Pascal Thubert, Teco Boot, Jim Bound, and Ben McCarthy. Problem statement and requirements for MANEMO. Internet-Draft draft-wakikawa-manemo-problem-statement-01.txt, IETF Secretariat, July 2007. 2

Ryuji Wakikawa, Vijay Devarapalli, George Tsirtsis, Thierry Ernst, and Kenichi Nagami. Multiple care-of addresses registration. RFC 5648, RFC Editor, October 2009. 22

Helen J. Wang, Randy H. Katz, and Jochen Giese. Policy-enabled handoffs across heterogeneous wireless networks. In Ramón Cáceres, editor, *WMCSA 1999, 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 51–60. IEEE Computer Society, February 1999. ISBN 0-7695-0025-0. doi: 10.1109/MCSA.1999.749277. 33, 34, 47

Margaret Wasserman and Pierrick Seite. Current practices for multiple interface hosts. Internet-Draft draft-ietf-mif-current-practices-07.txt, IETF Secretariat, February 2011. xviii, 5, 19, 62

Jolyon White, Guillaume Jourjon, Thierry Rakotoarivelo, and Max Ott. Measurement architectures for network experiments with disconnected mobile nodes. In Anastasius Gavras, Nguyen Huu Thanh, and Jeff Chase, editors, *TridentCom 2010, 6th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. ICST, Springer-Verlag Berlin, May 2010. xxiii, 9, 45, 100, 103, 189

Jörg Widmer. *Equation-Based Congestion Control for Unicast and Multicast Data Streams*. PhD thesis, University of Mannheim, May 2003. xxi, 15, 75

Bernhard Wiegel, Yvonne Gunter, and Hans P. Grossmann. Cross-layer design for packet routing in vehicular ad hoc networks. In Greg Bottomley and Robert Heath, editors, *VTC 2007-Fall, 66th IEEE Vehicular Technology Conference*, pages 2169–2173. IEEE Vehicular Technology Society, IEEE Computer Society, September 2007. doi: 10.1109/VETECF.2007.455. 25

Ashton L. Wilson, Andrew Lenaghan, and Ron Malyan. Optimising wireless access network selection to maintain QoS in heterogeneous wireless environments. In Leif Heinzl and Neeli Prasad, editors, *WPMC 2005, 8th International Symposium on Wireless Personal Multimedia Communications*, pages 1236–1240. NICT, September 2005. 33, 34, 35, 47

Stefan Winkler and Praveen Mohandas. The evolution of video quality measurement: From PSNR to hybrid metrics. *IEEE Transactions on Broadcasting*, 54(3):660–668, September 2008. ISSN 0018-9316. doi: 10.1109/TBC.2008.2000733. 41

Bo Xing and Nalini Venkatasubramanian. Multi-constraint dynamic access selection in always best connected networks. In Suresh Singh and Philippe Bonnet, editors, *MobiQuitous 2005, 2nd Annual International Conference on Mobile and Ubiquitous*, pages 56–64. IEEE Computer Society, ACM SIGMOBILE, IEEE Computer Society, July 2005. doi: 10.1109/MOBIQUITOUS.2005.39. 33, 34

George Xylomenos, George C. Polyzo, Petri Mähönen, and Mika Saaranen. TCP performance issues over wireless links. *IEEE Communications Magazine*, 39(4):52–58, April 2001. ISSN 0163-6804. doi: 10.1109/35.917504. xv, xviii, 1, 18, 55

Tara A. Yahiya and Hakima Chaouchi. An optimized handover decision for heterogenous wireless networks. In Juan-Carlos Cano and Richard W. Pazzi, editors, *PM2HW2N 2009, 4th ACM workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, pages 137–142. ACM SIGSIM, ACM, October 2009. ISBN 978-1-60558-621-2. doi: 10.1145/1641913.1641933. 33, 35

Xiaohuan Yan, Y. Ahmet Şekercioğlu, and Sathya Narayanan. A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks. *Computer Networks*, 54(11):1848–1863, August 2010. ISSN 1389-1286. doi: 10.1016/j.comnet.2010.02.006. 32

Kun Yang and Alex Galis. Policy-driven mobile agents for context-aware service in next generation networks. In Roch Glitho and Thomas Magedanz, editors, *MATA 2003, 5th International Workshop on Mobile Agents for Telecommunication Applications*, volume 2881 of *Lecture Notes in Computer Science*, pages 111–120. Springer-Verlag Berlin, October 2003. doi: 10.1007/978-3-540-39646-8_11. 34

Jun Yao, Salil S. Kanhere, and Mahbub Hassan. An empirical study of bandwidth predictability in mobile computing. In Roger Karrer and Lin Zhong, editors, *WiNTECH 2008, 3rd ACM international workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization*, pages 11–18. ACM SIGMOBILE, ACM, September 2008. ISBN 978-1-60558-187-3. doi: 10.1145/1410077.1410081. 34

Jun Yao, Salil S. Kanhere, and Mahbub Hassan. Geo-intelligent traffic scheduling for multi-homed on-board networks. In Jörg Ott and Kun Tan, editors, *MobiArch 2009, 4th International Workshop on Mobility in the Evolving Internet Architecture*. ACM SIGMOBILE, ACM, June 2009. ISBN 978-1-60558-688-5/09/06. 33, 34, 35, 63

Alper E. Yegin, Mohan Parthasarathy, and Carl Williams. Mobile IPv6 neighborhood routing for fast handoff. Internet-Draft draft-yegin-mobileip-nrouting-01.txt, IETF Secretariat, November 2000. 22

Mika Ylianttila, M. Pande, Juha-Pekka Mäkelä, and Petri Mähönen. Optimization scheme for mobile users performing vertical handoffs between IEEE 802.11 and GPRS/EDGE networks. In Arthur Henley and Booker Tyron, editors, *GLOBECOM 2001, IEEE Global Telecommunications Conference*, volume 6, pages 3439–3443. IEEE Communications Society, November 2001. ISBN 0-7803-7206-9. doi: 10.1109/GLOCOM.2001.966320. 26, 33

Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, and Jaakko Laine. Dynamic network interface selection in multihomed mobile hosts. In *HICSS-36, 36th Annual Hawaii International Conference on System Sciences*, January 2003. doi: 10.1109/HICSS.2003.1174876. 35

Tjalling J. Ypma. Historical development of the Newton–Raphson method. *SIAM Review*, 37(4):531–551, December 1995. ISSN 0036-1445. doi: 10.1137/1037125. 87

Vassilis E. Zafeiris and E. A. Giakoumakis. Mobile agents for flow scheduling support in multihomed mobile hosts. In Xi Zhang, editor, *IWCMC 2008, International Wireless Communications and Mobile Computing Conference*, pages 261–266. IEEE, August 2008. ISBN 978-1-4244-2201-2. doi: 10.1109/IWCMC.2008.46. 33, 34, 35

Matt Zekauskas. Network performance measurement tools: An Internet2 cookbook. Technical report, Internet2, 2005. 43, 56, 189

Lei Zhang, Patrick Sénac, Emmanuel Lochin, and Michel Diaz. Cross-layer based congestion control for WLANs. In Lionel Ni and Jiannong Cao, editors, *QShine 2008, 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pages 1–7. ACM SIGSIMM, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), June 2008a. ISBN 978-963-9799-26-4. 27

Lei Zhang, Patrick Sénac, Emmanuel Lochin, and Michel Diaz. Mobile TFRC: A congestion control for WLANs. In Giuseppe Anastasi and Nalini Venkatasubramanian, editors, *WoWMoM 2008, 9th International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–4. IEEE, June 2008b. ISBN 978-1-4244-2099-5. doi: 10.1109/WOWMOM.2008.4594867. 18, 27

Lixia Zhang, David Meyer, and Kevin Fall. Report from the IAB workshop on routing and addressing. RFC 4984, RFC Editor, September 2007. 19

Xinhua Zhao, Claude Castelluccia, and Mary Baker. Flexible network support for mobility. In William P. Osborne and Dhawal Moghe, editors, *MobiCom 1998, 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 145–156. IEEE Computer Society, ACM SIGCOMM, ACM SIGMOBILE, ACM, October 1998. ISBN 1-58113-035-X. doi: 10.1145/288235.288274. 35

Bosheng Zhou, Alan Marshall, Jieyi Wu, Tsung H. Lee, and Jiakang Liu. A cross-layer route discovery framework for mobile ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, 2005(5):645–660, October 2005. ISSN 1687-1472. doi: 10.1155/WCN.2005.645. 25

Hubert Zimmermann. OSI reference model—the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432, 1980. xv, 1

Appendix A

Summary of Contributions

A.1 Publications

Olivier Mehani and Roksana Boreli. Adapting TFRC to mobile networks with frequent disconnections. In Keith W. Ross and Leandros Tassiulas, editors, *CoNEXT 2008, 4th ACM International Conference on emerging Networking EXperiments and Technologies, Student Workshop*. ACM SIGCOMM, ACM, December 2008. ISBN 978-1-60558-210-8. doi: 10.1145/1544012.1544049.

Olivier Mehani, Roksana Boreli, and Thierry Ernst. Analysis of TFRC in disconnected scenarios and performance improvements with Freeze-DCCP. In Jörg Ott and Kun Tan, editors, *MobiArch 2009, 4th International Workshop on Mobility in the Evolving Internet Architecture*. ACM SIGMOBILE, ACM, June 2009a. ISBN 978-1-60558-688-5/09/06.

Olivier Mehani, Roksana Boreli, Guillaume Jourjon, and Thierry Ernst. Mobile multimedia streaming improvements with Freeze-DCCP. In Romit R. Choudhury and Henrik Lundgren, editors, *MobiCom 2010, 16th Annual International Conference on Mobile Computing and Networking, Demonstration Session*. ACM SIGMOBILE, September 2010.

Olivier Mehani, Roksana Boreli, and Thierry Ernst. Context-adaptive vehicular network optimization. In Marion Berbineau, Makoto Itami, and GuangJun Wen, editors, *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*, pages 186–191. IEEE Computer Society, October 2009b. ISBN 1-4244-1178-5.

Olivier Mehani, Roksana Boreli, Michael Maher, and Thierry Ernst. User- and application-centric multihomed flow management. In Tom Pfeifer and Anura Jayasumana, editors, *LCN 2011, 36th IEEE Conference on Local Computer Networks*, pages 26–34. IEEE Computer Society, IEEE Computer Society, October 2011a. ISBN 978-1-61284-928-7.

Olivier Mehani, Guillaume Jourjon, Jolyon White, Thierry Rakotoarivelo, Roksana Boreli, and Thierry Ernst. Characterisation of the effect of a measurement library on the performance of instrumented tools. Technical Report 4879, NICTA, May 2011b.

Manabu Tsukada, Olivier Mehani, and Thierry Ernst. Simultaneous usage of NEMO and MANET for vehicular communication. In Miguel P. de Leon, editor, *TridentCom 2008, 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), March 2008. ISBN 978-963-9799-24-0. 35

José Santa, Manabu Tsukada, Thierry Ernst, Olivier Mehani, and Antonio F. Gómez-Skarmeta. Assessment of VANET multi-hop routing over an experimental platform. *International Journal of Internet Protocol Technology*, 4(3):158–172, September 2009. ISSN 1743-8209. doi: 10.1504/IJIPT.2009.028655.

Manabu Tsukada, José Santa, Olivier Mehani, Yacine Khaled, and Thierry Ernst. Design and experimental evaluation of a vehicular network based on NEMO and MANET. *EURASIP Journal on Advances in Signal Processing*, 2010:1–18, September 2010. ISSN 1687-6180. doi: 10.1155/2010/656407. 44

Terence Chen, Olivier Mehani, and Roksana Boreli. Trusted routing for VANET. In Marion Berbineau, Makoto Itami, and GuangJun Wen, editors, *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*, pages 647–652. IEEE Computer Society, October 2009. ISBN 1-4244-1178-5.

A.2 Software

A.2.1 Freeze-TCP

- *ns*-2 module: http://www.nicta.com.au/people/mehanio/freezetcp#nsfreezetcp
- Linux-2.6 implementation: http://github.com/shtrom/linux-2.6/tree/freezetcp/

A.2.2 Freeze-DCCP/TFRC

- *ns*-2 patch: http://www.nicta.com.au/people/mehanio/freezedccp#nsfreezedccp
- Linux-2.6 implementation: http://github.com/shtrom/linux-2.6/tree/freezedccp/

A.2.3 Additions to OML

- OMF Measurement Library (OML)-enabled Iperf: http://mytestbed.net/projects/iperf/repository/show?rev=oml%2Fmaster
- Various contributions to the OML codebase (liboml2 and filters http://mytestbed.net/projects/oml/repository; oml-apps http://mytestbed.net/projects/omlapp/repository)

A.2.4 Miscellaneous

- IPv6 support patch integrated in vanilla GPSd: http://lists.berlios.de/pipermail/gpsd-commit-watch/2010-January/003759.html
- Various *ns*-2 patches and featureset updates: http://www.nicta.com.au/people/mehanio/nsmisc

Appendix B

Presentation Slides

Contributions to Mechanisms for Adaptive Use of Mobile Network Resources

Olivier Mehani

Supervisors: Roksana Boreli, Thierry Ernst,
Arnaud de La Fortelle, Aruna Seneviratne

14 December 2011

Notes

Notes

Introduction

A wireless world

- Prevalence of wireless access
  - Unlicensed: Wi-Fi, Bluetooth, ...
  - Licensed: 2–4G, WiMAX, ...
  - Built-in support for multiple technologies
- New connectivity modes
  - MANETs, VANETs, ...
  - DTNs, UPNs, ...
- "Always best connected" devices [References on slide 38]
- Increase computational power in mobile devices
- Emerging uses
  - Multimedia
    - VoIP, Video streaming, Video conferencing, ...
  - Mobility → ITS
    - Route planning, safety, traffic and fleet management, infotainment

Notes

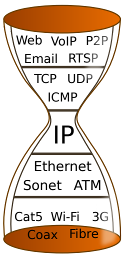
Introduction

Problem statement...

How to enable mobile communicating peers to make the best use of the network resources when they are available, and degrade gracefully when they are scarce?

Notes

Introduction
State of the stack^Wart



Based on image by Xander89

- Mobility
  - Proposals at every single layer
  - Ubiquitous network layer
    - *MIPv6*
- Cross-layer designs
  - Information/control over multiple layers
  - In-stack too specific
  - *Vertical control plane*
    - Standards: IEEE 802.21, ISO CALM manager, ETSI ITSC management
- Network selection
  - Unclear *which metrics are relevant*
    - S(I)NR, network QoS, prioritising (*e.g.*, Wi-Fi with 3G fallback), ...
  - Rare *fine grained management of multiple interfaces*

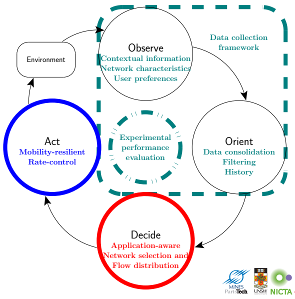
[Selected references on slide 38]

Notes

Introduction
Problem statement... and how to address it

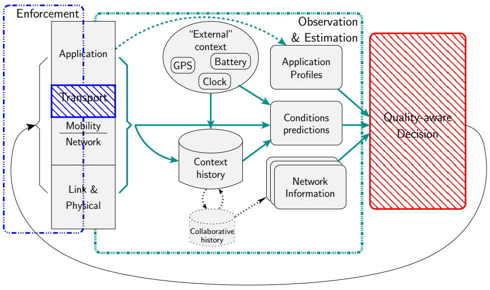
How to enable mobile communicating peers to make the best use of the network resources when they are available, and degrade gracefully when they are scarce?

- Research directions
  - Network selection
  - Adaptation to changes
  - Provide incremental modifications
- Approach: OODA loop
- Contribution axes
  - Optimisation of networks selection and use
  - Improvement of rate control mechanism for mobility
  - Study of measurement platforms and tools

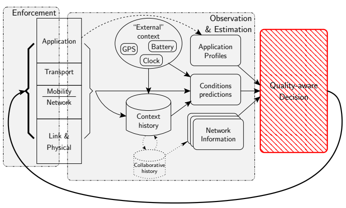


Notes

Introduction
Overarching control framework



Notes

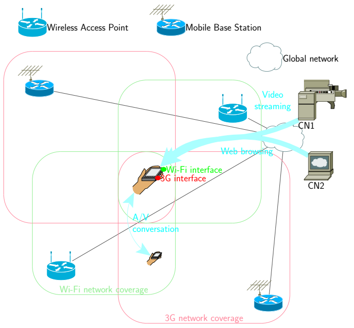


Notes

Multi-layer Optimisation of Network Choice and Usage

Problem of a multihomed mobile node: Mix and match?

- Multiple networks, interfaces and flows (of different types)
- How to decide
  - *Which interface(s) to use?*
  - *Which network(s) to connect to?* (*e.g.*, BS or ESS)
  - *How to distribute the flows?*
- To optimise. . . *what?*
  - raw QoS (*e.g.*, goodput or delays)?
- *Multihomed Flow Management problem*

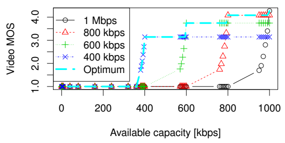


Notes

Multi-layer Optimisation of Network Choice and Usage

Problem of a multihomed mobile node: Privileging users' perceptions and expectations

- User experiences the application's output
  - QoS *only* directly relevant to the application
    - Adjustable parameters
    - Optional requirements
    - Non-linear QoE/QoS relation (*e.g.*, H.264) [ITU E-Model on slide 42]



- Flat battery the worst experience
- User's wallet not a bottomless bag
- *Conflicting goals*
  - Need for tradeoffs

Notes

Multi-layer Optimisation of Network Choice and Usage

Problem of a multihomed mobile node: Formalisation

- Quality-aware Multihomed Flow Management
  - Maximise application quality [UML on slide 41]
  - Reduce costs
    - Energy consumption
    - Access price
  - Decision scope
    - (De)activate interfaces
    - Select most appropriate networks
    - Distribute flows
    - Adjust stack parameters (*e.g.*, application or transport)
- Constrained optimisation model [Notations on slide 43]
  - MiniZinc language
  - Branch-and-bound search
  - Optimal solution

Notes

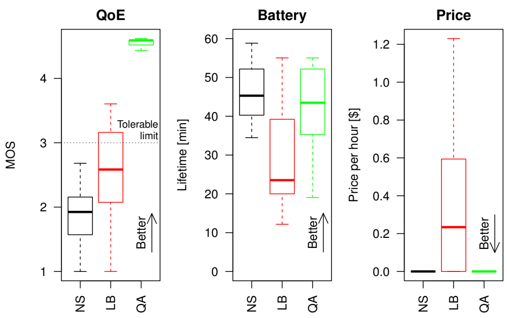
Multi-layer Optimisation of Network Choice and Usage

Evaluation and comparison: Techniques and scenarios

- Comparison to more common techniques
  - QA Quality-aware Multihomed Flow Management [Objective function on slide 44]
  - NS Single network/interface selection (*e.g.*, iPhones) [Objective function on slide 45]
  - LB Load balancing on each interface's best network [Objective function on slide 46]
- Two types of scenarios
  - Smart-phone example: Single Wi-Fi and 3G interfaces, random networks, fixed demand (2 VoIP and video flows, 3 web sessions)
  - More generic scenarios: Interfaces, networks and flows chosen randomly

Notes

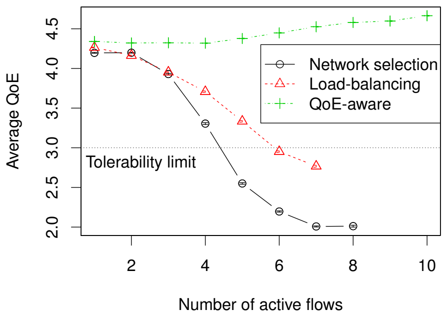
Multi-layer Optimisation of Network Choice and Usage
Evaluation and comparison: Smart-phone example



[Data sources on slide 47] [Approach and data quality on slide 48]

Notes

Multi-layer Optimisation of Network Choice and Usage
Evaluation and comparison: Generic scenarios



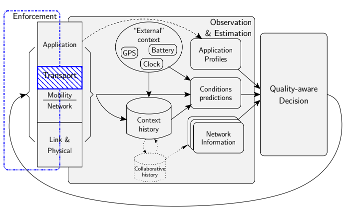
[Battery and price on slide 49] [Medians on slide 50] [Data sources on slide 47]
[Approach and data quality on slide 48]

Notes

Multi-layer Optimisation of Network Choice and Usage
Summary and future work

- Summary
  - Mobile-centric model of multihoming
    - Constrained formulation
  - QoE-aware Multihomed Flow Management
    - Evaluation
    - Real data from QoS testbed
- Future work
  - Global stability
  - Finer-grained routes
  - Extension to NEMOs
  - Actual implementation
    - Linear programming formulation
    - Remove simplificating assumptions (*e.g.*, direction of flows, pricing)
    - Prioritisation weights from user

Notes



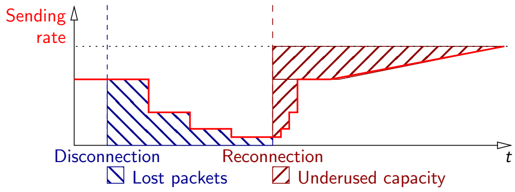
Notes

Mobility-aware rate control for transports
Problem: Classical congestion control assumptions broken by mobility

- *TCP-Friendly Rate Control* (TFRC) [References on slide 52]
  - Rate-based congestion control mechanism
    - TCP-fair congestion control
    - Uses packets losses p and RTT R
    - Well adapted to real-time streaming
  - Used with *Datagram Congestion Control Protocol* (DCCP)
    - Unreliable datagrams
    - Real-time traffic over shared networks
- Problems with mobility
  - Losses during hand-off period force a rate reduction
  - Poor adaptability to new network characteristics
- *How much resources are wasted?*
- *How not to?*

Notes

Mobility-aware rate control for transports
TFRC during handovers: Modelling losses and wasted capacity



- Sending rate
- Lost packets during disconnection
- "Wasted" capacity after reconnection
- Additional "wasted" capacity on higher capacity networks

[Formulas on slide 53]

Notes

Mobility-aware rate control for transports
TFRC during handovers: Analytically derived possible performance improvements

from \ to	UMTS	802.16	802.11 b	802.11 g
Packet losses				
UMTS	3×10^2			
802.16	2×10^3			
802.11b	1×10^3			
802.11g	3×10^3			
Unused capacity [500 B packets]				
UMTS	0	$\ddots$	$\vdots$	
802.16	0	2×10^2–8×10^4		
802.11b	$\vdots$	$\ddots$		$\ddots$
802.11g	0	$\cdots$	0	

[Real numbers on slide 54]

Notes

Mobility-aware rate control for transports
Solution: Temporarily "freezing" the transport to avoid losses

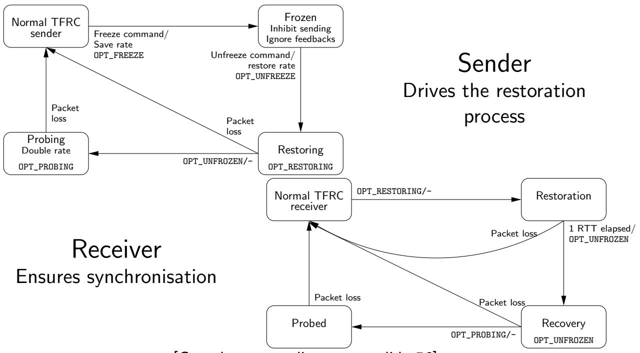
- Freeze-DCCP/TFRC
  - Sender/receiver cooperation
    - DCCP-level options
  - New states supporting
    1. rate restoration
    2. path probing
- Related work: Freeze-TCP [References on slide 55]
  - Predictable disconnections at receiver
  - Suspend TCP traffic
  - Restore rate on reconnection
- Better support for mobility handoffs

  sender-based freezing for mobile senders
  slow-start-like probing for higher new capacities

Notes

Mobility-aware rate control for transports

Mobility-Aware extension to TFRC: Additional states and options to support freezing

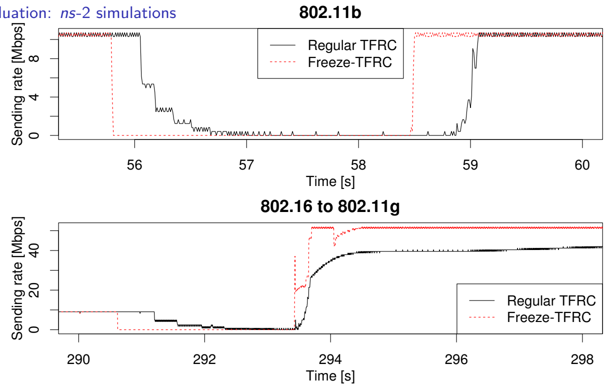


[Complete state diagram on slide 56]

Notes

Mobility-aware rate control for transports

Evaluation: *ns*-2 simulations



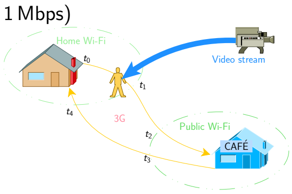
[Numbers on slide 57]

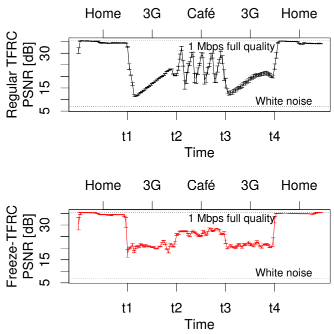
Notes

Mobility-aware rate control for transports

Evaluation: Experiments with emulated handovers

- Video streaming (H.264, 1 Mbps)





- QoE metric: PSNR
- Linux kernel code
- Emulated links and handovers [References on slide 55]
  - Home: 1 Mbps, 52 ms
  - 3G: 500 kbps, 250 ms
  - Café: 700 kbps, 70 ms

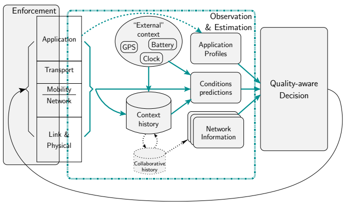
Notes

Mobility-aware rate control for transports

Summary and future work

- Summary
  - Model TFRC in vertical handovers
  - Freeze-TFRC protocol within DCCP
    - *ns*-2
    - Linux
    - Evaluation
- Future work
  - Robustness of state machine
  - Decouple freezing and probing to cater for “make before break”
  - Stopping criteria for probing
    - Information from decision framework

Notes



Notes

Accuracy of a Measurement Instrumentation Library

Problem: Obtaining accurate measurements

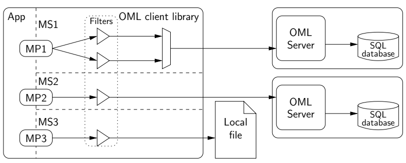
- Network measurements needed at every step
  - design based on observations
  - monitoring of the world
  - experimentation to evaluate performance
- Requirements for network measurement tools
  - generic multiple different experiments
  - validated confidence in the measurements
  - extensible as many variables as possible
- *Needed for the information reporting loop of the framework*

Notes

Accuracy of a Measurement Instrumentation Library

Problem: Obtaining accurate measurements

- OMF Measurement Library (OML) [References on slide 62]
  - Open Source C library (MIT licensed)
  - Timestamped samples
  - Unified output format (SQL databases)
  - Instrumentation of *already existing applications*
  - In-line filtering and aggregation
  - Domain-free (*cf.* SNMP for network, DTrace for systems)



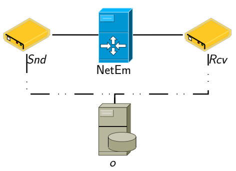
- *Are reports using OML trustworthy?*

Notes

Accuracy of a Measurement Instrumentation Library

OML's Impact on Instrumented Applications: Experimental setup

- Instrumented measurement tools
  - Active: `iperf`(1)
  - Passive: `pcap`(3)-based packet capture
  - System load
- Generic experiments
- Various factors
  - `iperf`(1): traffic rate, OML support, threads [Some results on slide 63]
  - `pcap`(3): traffic rate, OML support [Some results on slide 64]
- Statistical tests
  - (PERM)ANOVA
  - Data usability: Standard error, independence, normality, homoskedasticity



Notes

Accuracy of a Measurement Instrumentation Library
Summary and future work

- Summary
  - First evaluation of OML's operation ranges
  - Non-threaded reporting performs equally to a threaded application
  - Bottleneck in passive measurement beyond 50 Mbps when all packets are reported
  - Seems adequate for the proposed framework
- Future work
  - Instrument more applications
  - Remove bottleneck

Notes

Notes

Summary
Contributions

- Models, validation and evaluation
  - Multihomed device
  - Quality-based decision
  - TFRC in handovers
- Protocols and software
  - Freeze-DCCP/TFRC (Linux, *ns*-2)
  - Freeze-TCP (Linux, *ns*-2)
  - Ported other *ns*-2 patches (DCCP, MobiWan)
  - Additions to OML codebase
- Experimental evaluation
  - Freeze-DCCP/TFRC
  - OML

[Publications on slide 36]

Notes

Summary
Future work and perspectives

- Future Work
  - Framework implementation
  - Applicability of the results
    - Relation to ITS standards?
- Perspectives
  - Evolution-limiting factor
    - Direct application access to `socket`(2) interface
    - Higher-level interface needed (*e.g.*, hide network names, provide service exposure and discovery or perform local and remote firewall configuration)
  - Internet Hourglass' waist too narrow, transport too deep
    - Network layer should expose more information (*e.g.*, detected paths or congestion)
    - Transport should be split: Per peer path-to-host congestion management (channel), per channel packet scheduling and high level semantics (transport)

Notes

Notes

Questions?

Thanks

Notes

Backup

Notes

Backup: Publications

Notes

- QA-MFM

  - Olivier Mehani, Roksana Boreli, and Thierry Ernst. "Context-Adaptive Vehicular Network Optimization". In: *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*. Ed. by Marion Berbineau, Makoto Itami, and GuangJun Wen. Lille, France: IEEE Computer Society, Oct. 2009, pp. 186–191. ISBN: 1-4244-1178-5
  - Olivier Mehani, Roksana Boreli, Michael Maher, and Thierry Ernst. "User- and Application-Centric Multihomed Flow Management". In: *LCN 2011, 36th IEEE Conference on Local Computer Networks*. Ed. by Tom Pfeifer and Anura Jayasumana. IEEE Computer Society. IEEE Computer Society, Oct. 2011, pp. 26–34

Backup: Publications

- Freeze-TFRC
  - Olivier Mehani and Roksana Boreli. "Adapting TFRC to Mobile Networks with Frequent Disconnections". In: *CoNEXT 2008, 4th ACM International Conference on emerging Networking EXperiments and Technologies, Student Workshop*. Ed. by Keith W. Ross and Leandros Tassiulas. ACM SIGCOMM. Madrid, Spain: ACM, Dec. 2008. ISBN: 978-1-60558-210-8. DOI: 10.1145/1544012.1544049
  - Olivier Mehani, Roksana Boreli, and Thierry Ernst. "Analysis of TFRC in Disconnected Scenarios and Performance Improvements with Freeze-DCCP". In: *MobiArch 2009, 4th International Workshop on Mobility in the Evolving Internet Architecture*. Ed. by Jörg Ott and Kun Tan. ACM SIGMOBILE. Kraków, Poland: ACM, June 2009. ISBN: 978-1-60558-688-5/09/06
  - Olivier Mehani, Roksana Boreli, Guillaume Jourjon, and Thierry Ernst. "Mobile Multimedia Streaming Improvements with Freeze-DCCP". In: *MobiCom 2010, 16th Annual International Conference on Mobile Computing and Networking, Demonstration Session*. Ed. by Romit R. Choudhury and Henrik Lundgren. ACM SIGMOBILE. Chicago, IL, USA, Sept. 2010

Notes

Backup: Publications

- OML
  - Olivier Mehani et al. *Characterisation of the Effect of a Measurement Library on the Performance of Instrumented Tools*. Tech. rep. 4879. NICTA, May 2011

Notes

Backup: Publications

- Others
  - Manabu Tsukada, Olivier Mehani, and Thierry Ernst. "Simultaneous Usage of NEMO and MANET for Vehicular Communication". In: *TridentCom 2008, 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*. Ed. by Miguel P. de Leon. Innsbruck, Austria: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Mar. 2008. ISBN: 978-963-9799-24-0
  - Terence Chen, Olivier Mehani, and Roksana Boreli. "Trusted Routing for VANET". In: *ITST 2009, 9th International Conference on Intelligent Transport Systems Telecommunications*. Ed. by Marion Berbineau, Makoto Itami, and GuangJun Wen. Lille, France: IEEE Computer Society, Oct. 2009, pp. 647–652. ISBN: 1-4244-1178-5
  - José Santa et al. "Assessment of VANET Multi-hop Routing over an Experimental Platform". In: *International Journal of Internet Protocol Technology* 4.3 (Sept. 2009), pp. 158–172. ISSN: 1743-8209. DOI: 10.1504/IJIPT.2009.028655
  - Manabu Tsukada et al. "Design and Experimental Evaluation of a Vehicular Network Based on NEMO and MANET". In: *EURASIP Journal on Advances in Signal Processing* 2010 (Sept. 2010). Ed. by Hossein Pishro-Nik, Shahrokh Valaee, and Maziar Nekovee, pp. 1–18. DOI: 10.1155/2010/656407

Notes

Notes

Backup: Selected references

ABC Eva Gustafsson and Annika Jonsson. "Always Best Connected". In: *IEEE Wireless Communications* 10.1 (Feb. 2003). Ed. by Michele Zorzi, Abbas Jamalipour, and Masami Yabusaki, pp. 49–55. ISSN: 1536-1284. DOI: 10.1109/MWC.2003.1182111

Mobility Fawad Nazir and Aruna Seneviratne. "Towards Mobility Enabled Protocol Stack for Future Wireless Network". In: *Ubiquitous Computing and Communication Journal* 2.4 (Aug. 2007). Ed. by Usman Tariq

Cross-Layer Sanjay Shakkottai, Theodore S. Rappaport, and Peter C. Karlsson. "Cross-Layer Design for Wireless Networks". In: *IEEE Communications Magazine* 41.10 (Oct. 2003). Ed. by Wojciech Kabacinski, Chin-Tau Lea, and Guoliang Xue, pp. 74–80. ISSN: 0163-6804. DOI: 10.1109/MCOM.2003.1235598

Risks Vikas Kawadia and P. R. Kumar. "A Cautionary Perspective on Cross-layer Design". In: *IEEE Wireless Communications* 12.1 (Feb. 2005). Ed. by Michele Zorzi, pp. 3–11. ISSN: 1536-1284. DOI: 10.1109/MWC.2005.1404568

Decision Xiaohuan Yan, Y. Ahmet Şekercioğlu, and Sathya Narayanan. "A Survey of Vertical Handover Decision Algorithms in Fourth Generation Heterogeneous Wireless Networks". In: *Computer Networks* 54.11 (Aug. 2010). Ed. by Ian F. Akyildiz and Harry Rudin, pp. 1848–1863. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2010.02.006

Notes

Notes

Backup: Multihomed Flow Management

References

Datasets Henrik Petander. "Energy-aware Network Selection Using Traffic Estimation". In: *MICNET 2009, 1st ACM workshop on Mobile Internet through Cellular Networks*. Ed. by Songwu Lu and Hewu Li. ACM SIGMOBILE. Beijing, China: ACM, Sept. 2009, pp. 55–60. ISBN: 978-1-60558-753-0. DOI: 10.1145/1614255.1614268

MOS . *Methods for Subjective Determination of Transmission Quality*. ITU-T SG12. Aug. 1996

VoIP . *The E-Model, a Computational Model for Use in Transmission Planning*. ITU-T SG12. Mar. 2005

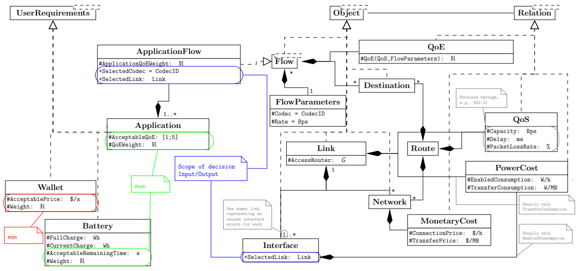
Video . *Opinion Model for Video-Telephony Applications*. ITU-T SG12. Apr. 2007

Web . *Estimating End-to-End Performance in IP Networks for Data Applications*. ITU-T SG12. May 2006

Notes

Backup: Multihomed Flow Management

UML model



Constraints: Associations' multiplicity is 1 unless otherwise stated.

$$\sum_{f \in F | f.\texttt{SelectedLink}=l} \text{FlowParameters}(f.\texttt{SelectedCodec}).\texttt{Rate} \leq l.\texttt{Capacity} \; \forall l \in L$$

$$\sum_{f \in F | Network(f.\texttt{SelectedLink})=n, Destination(f)=d} \text{FlowParameters}(f.\texttt{SelectedCodec}).\texttt{Rate} \leq n.\texttt{Capacity} \; \forall n \in N, d \in D$$

Notes

Backup: Multihomed Flow Management

ITU-T's QoE

VoIP $R = 93.193 - I_s - I_d - I_{e-eff}$

Video $V_q = 1 + I_{coding} \exp\left(\frac{P_{plV}}{D_{PplV}}\right)$ (linear combination for A/V)

Web $MOS_{\text{web}} = 5 + 4 \cdot \frac{\ln(WeightedST) - \ln(Min)}{\ln(Min) - \ln(Max)}$,
$WeightedST = 0.98 \cdot T_3 + 1.76 \cdot T_4$ (discarding search phase)

[References on slide 40]

Notes

Backup: Multihomed Flow Management

Notations

	Set of networks N
$\text{None} \in N$	null network to represent unassociated interfaces
	Set of interfaces I
$\vec{A}, \lvert\vec{A}\rvert = \lvert I\rvert$	network association vector where $A_i \in N, \forall i \in I$
	Set of links $L \subseteq I \times N$
$QoS(l)$	achievable QoS achievable on link $l \in L$
$Pw(l)$	power consumption of link l
$Pr(l)$	access price of link l
	QoS tuple $q = \langle c, r, e, s, \ldots \rangle$
$C(q) = c$	available capacity
$R(q) = r$	round-trip time
e	link error rate
s	security index
$\ldots$	other metrics relevant to an application
	Set of flows F
$\vec{D}, \lvert\vec{D}\rvert = \lvert F\rvert$	flow distribution vector where $D_f \in L, \forall f \in F$
$\vec{p}, \lvert\vec{p}\rvert = \lvert F\rvert$	application-specific parameters (p_f for flow f)
$Q(f, p_f, q_f)$	quality profile of flow $f \in F$ under QoS q_f
$q_{\text{req}}(f, p_f)$	min. required QoS to maximise $Q(f, p_f, q_{\text{req}}(f, p_f))$

Notes

Backup: Multihomed Flow Management

Multihomed Flow Management objective

$$\max_{\vec{A},\vec{D},\vec{p}} \left(\sum_{f\in F} W_f Q(f, p_f, q_{\text{req}}(f, p_f)) - W_b \sum_{i \in I} Pw(l_i) - W_p \sum_{i\in I} Pr(l_i) \right) \quad (1)$$

$$\begin{cases} \forall f \in F, \exists i \in I \quad A_i \neq \text{None} \wedge D_f = l_i, & (2a) \\ \forall i \in I \quad \sum_{f \in F \mid D_f = l_i} C(q_{\text{req}}(f, p_f)) \leq C(QoS(l_i)) & (2b) \end{cases}$$

[Notations on slide 43]

Notes

Backup: Multihomed Flow Management

Network Selection objective

$$\max_{\vec{A}} \sum_{i \in I} C(l_i) \quad (3)$$
$$\text{s.t.} \begin{cases} \exists i \in I \quad A_i \neq \text{None} \\ \forall j \in I - \{i\} \quad A_j = \text{None} \end{cases}$$

[Notations on slide 43]

Notes

Backup: Multihomed Flow Management
Load Balancing objective

$$Lr(l) = \sum_{f \in F | D_f = l} C(q_{\text{req}}(p_f))/C(l)$$

$$F_r = \frac{\left(\sum_{i \in I} Lr(l_i)\right)^2}{|I| \sum_{i \in I} Lr(l_i)^2} \quad (4)$$

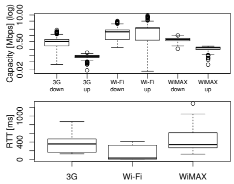
$$\max_{\vec{A},\vec{D}} \left(W_c \sum_{i \in I} C(l_i) + W_f F_r \right) \quad (5)$$

[Notations on slide 43]

Notes

Backup: Multihomed Flow Management
Supporting data

- QoS measurement testbed
  - Wi-Fi, WiMAX, 3G (Australia, Germany)
  - Know measurement servers (Australia, France)
  - Sep.–Nov. 2010

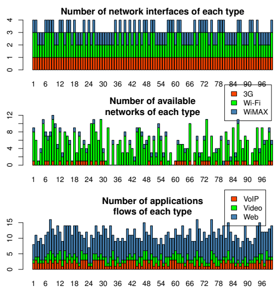


- Quality profiles
  - MOS from ITU-T's objective E-Model [Formulas on slide 42]
  - VoIP, video conferencing, web browsing
  - Easily extended given similarly formulated objective profiles
    - Other interactive applications
    - Non-interactive applications with evaluable performance
- Battery consumption and web usage data from Petander (2009) [References on slide 40]
- Access prices surveyed from Australian operators in Dec. 2010

Notes

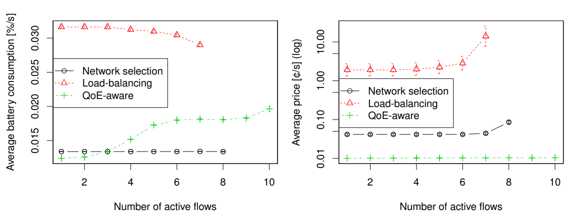
Backup: Multihomed Flow Management
Approach and data quality

- Scenarios
  - Smart-phone (subset): 57
  - Synthetic (total): 95
- Run from 1 to n flows
  - Evaluate behaviours with increasing load
  - For 7 flows, usually not more than 20 s
    - Not quite real-time...
- Statistical significance of averages
  - At least 20 data points
  - Discarded otherwise



Notes

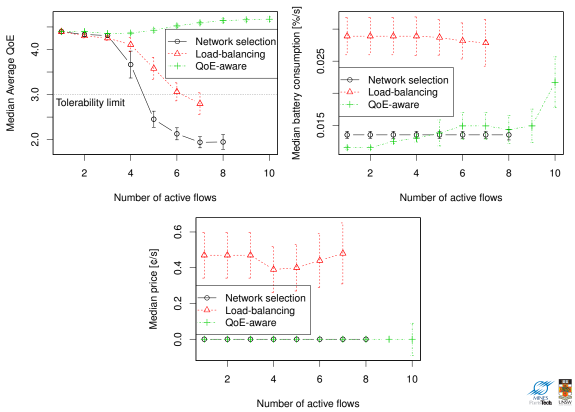
Backup: Multihomed Flow Management
Generic scenarios, battery and price results



Notes

Backup: Multihomed Flow Management

More results



Notes

Publications

Selected references

Multihomed Flow Management

Freeze-TFRC

OML

Notes

Backup: Freeze-TFRC

References

TCP Reno model Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. “Modeling TCP Throughput: A Simple Model and Its Empirical Validation”. In: *SIGCOMM Computer Communication Review* 28.4 (Oct. 1998). Ed. by Martha Steenstrup, pp. 303–314. ISSN: 0146-4833. DOI: 10.1145/285243.285291

- $X_{\mathrm{Bps}}(p, R) = \frac{s}{R\sqrt{\frac{4p}{3}}+t_{\mathrm{RTO}}\sqrt{\frac{27p}{8}}p(1+32p^2)}$

TFRC Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. *TCP Friendly Rate Control (TFRC): Protocol Specification.* RFC 5348. RFC Editor, Sept. 2008

DCCP Eddie Kohler, Mark Handley, and Sally Floyd. *Datagram Congestion Control Protocol (DCCP).* RFC 4340. RFC Editor, Mar. 2006

Mobility and adaptability Deepak Bansal, Hari Balakrishnan, Sally Floyd, and Scott Shenker. “Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms”. In: *SIGCOMM 2001, Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications.* Ed. by Rene Cruz and George Varghese. San Diego, CA, USA: ACM, Aug. 2001, pp. 263–274. ISBN: 1-58113-411-8. DOI: 10.1145/383059.383080

Notes

Backup: Freeze-TFRC

Modelling losses and capacity wastage

$$X^i = \begin{cases} \frac{X_d}{2^i} & \text{if } 0 \leq i < i_x, \\ \frac{s}{t_{\mathrm{mbi}}} & \text{otherwise,} \end{cases}$$

$$n_{\mathrm{lost}} = \begin{cases} \left\lfloor \frac{7}{8}\frac{t_D X^0}{s} \right\rfloor \\ \qquad (t_D \leq t^0_{\mathrm{RTO}}) \\ \left\lfloor \frac{7}{8}\frac{t^0_{\mathrm{RTO}} X^0}{s} + \sum_{i=1}^{i_D-1} \frac{t^i_{\mathrm{RTO}} X^i}{s} + \frac{t^{i_D}_{\mathrm{RTO}} X^{i_D}}{2s} \right\rfloor \\ \qquad \text{(otherwise)} \end{cases} \tag{6}$$

$$n_{\mathrm{wasted}} = \frac{1}{s}\left(t_{\mathrm{idle}} \cdot X_d + \sum_{i=0}^{n_{\mathrm{ss}}} R_{\mathrm{new}}\left(X_d - 2^i X_c\right)\right) \tag{7}$$

$$n'_{\mathrm{wasted}} = \frac{1}{s}(X_{\mathrm{max}} - X_d)(t_{\mathrm{idle}} + t_{\mathrm{ss}}) + \frac{R_{\mathrm{new}}}{s}\sum_{i=0}^{n_{\mathrm{grow}}}\left(X_{\mathrm{max}} - X^i\right) \tag{8}$$

Notes

Backup: Freeze-TFRC

TFRC during handovers: Analytically derived possible performance improvements

from \ to	UMTS	802.16	802.11 b	802.11 g
Packet losses				
UMTS	306	236	226	224
802.16	2760	2614	2614	2614
802.11b	1080	1078	1078	1078
802.11g	2909	2907	2907	2907
Unused capacity [500 B packets]				
UMTS	0	82938	263	109541
802.16	0	471	155	1029
802.11b	0	0	1085	54674
802.11g	0	0	0	4699

[Simulation results on slide 57] [Link characteristics on slide 58]

Notes

Backup: Freeze-TFRC

References

Freeze-TCP Tom Goff, James Moronski, Dhananjay S. Phatak, and Vipul Gupta. "Freeze-TCP: A True End-to-end TCP Enhancement Mechanism for Mobile Environments". In: *INFOCOM 2000, 19th Annual Joint Conference of the IEEE Computer and Communications Societies.* Ed. by Raphael Rom and Henning Schulzrinne. Vol. 3. Tel-Aviv, Israel: IEEE Computer Society, Mar. 2000, pp. 1537–1545. ISBN: 0-7803-5880-5. DOI: 10.1109/INFCOM.2000.832552

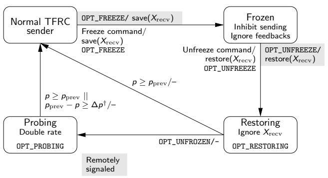
Wireless emulation Andrei Gurtov and Sally Floyd. "Modeling Wireless Links for Transport Protocols". In: *SIGCOMM Computer Communication Review* 34.2 (Apr. 2004). Ed. by John Wroclawski, pp. 85–96. ISSN: 0146-4833. DOI: 10.1145/997150.997159

Handover durations Jun S. Lee, Seok J. Koh, and Sang H. Kim. "Analysis of Handoff Delay for Mobile IPv6". In: *VTC2004-Fall, 60th IEEE Vehicular Technology Conference.* Ed. by Tien M. Nguyen. Vol. 4. IEEE Computer Society, Sept. 2004, 2967–2969 Vol. 4. ISBN: 0-7803-8521-7. DOI: 10.1109/VETECF.2004.1400604

Notes

Backup: Freeze-TFRC

Mobility-Aware extension to TFRC: Additional states and options to support freezing

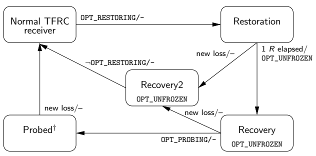


Sender

Drives the restoration process

Receiver

Ensures synchronisation



†When a packet is lost, the receiver computes and reports a p equivalent to the currently observed X_{recv}.

Notes

Backup: Freeze-TFRC

Evaluation: *ns-2* simulations

from \ to	UMTS	802.16	802.11 b	802.11 g
Packet losses (DCCP/TFRC only)				
UMTS	253.3	269.8	273.6	275.4
802.16	1732.3	1734.6	1734.6	1734.6
802.11b	856	855.5	855.3	855.3
802.11g	2470.9	2470.4	2470.2	2470.1
Unused capacity [500 B packets]				
UMTS	50.5	54018.05	2209.5	92156.1
—	13.4	3607.9	*9342.75*	89328.6
802.16	12.45	1827.95	603.05	4185.75
—	5	591.15	150.9	1520.35
802.11b	150.45	28314	2101.75	57970.65
—	0	15278	47.45	1045.05
802.11g	42.5	2104.3	943.4	4313
—	0	*7172.75*	46.5	188.45

[Analytical predictions on slide 54] [Fairness on slide 60] [Link characteristics on slide 58] [Handover durations on slide 59] [References on slide 55]

Notes

Backup: Freeze-TFRC
Commonly accepted link characteristics

Technology	Capacity [bps]	Delay [s]
UMTS	384 k	125 m
802.11b/g	11 M/54 M	10 m
802.16	9.5 M	40 m

[References on slide 55]

Notes

Backup: Freeze-TFRC
Handover durations

$$T_{\text{handoff}} = 2.5 + RTT_{\text{wireless}} + RTT_{\text{wired}}$$
$$= 2.6 + 2Delay_{\text{wireless}}$$

Destination network	T_{handoff} [s]
UMTS	2.85
802.16	2.68
802.11b/g	2.62

[References on slide 55] [Link characteristics on slide 58]

Notes

Backup: Freeze-TFRC
Fairness

- Single TCP flow from *AR* to *CN*
- Wait for settlement of rate after reconnection
- 100 s samples afterwards

from \ to	UMTS	802.16	802.11 b	802.11 g
UMTS	0.6	0.3	0.2	0.1
802.16	1.6	1.3	1.1	0.9
802.11b	1.3	1	0.9	0.7
802.11g	1.5	1.2	1	1.1

- Values in $[0.5, 2]$ considered "reasonably fair"
- Closely similar to DCCP/TFRC in the same conditions

[Link characteristics on slide 58] [Handover durations on slide 59]

Notes

Notes

Backup: OML

References

OML Jolyon White, Guillaume Jourjon, Thierry Rakotoarivelo, and Max Ott. “Measurement Architectures for Network Experiments with Disconnected Mobile Nodes”. In: *TridentCom 2010, 6th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*. Ed. by Anastasius Gavras, Nguyen Huu Thanh, and Jeff Chase. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. ICST. Berlin, Germany: Springer-Verlag Berlin, May 2010

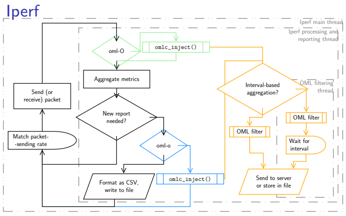
SNMP David Harrington, Randy Presuhn, and Bert Wijnen. *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*. RFC 3411. RFC Editor, Dec. 2002

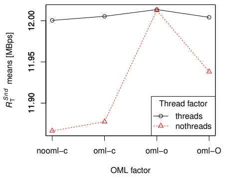
DTrace Bryan M. Cantrill, Michael W. Shapiro, and Adam H. Leventhal. “Dynamic Instrumentation of Production Systems”. In: *USENIX 2004*. Ed. by Andrea Arpaci-Dusseau and Remzi Arpaci-Dusseau. Boston, MA, USA: USENIX Association, June 2004, pp. 15–28

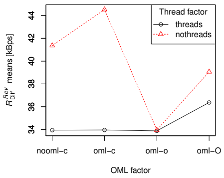
Notes

Backup: OML

Active measurement, Iperf







Actual sending rate at 95 Mbps

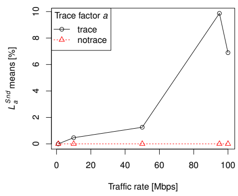
Difference in report at 50 Mbps

Notes

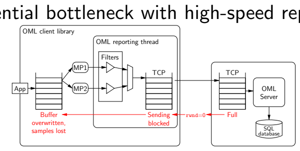
Backup: OML

Passive measurement, libpcap

- Number of unreported packets



- Hints at potential bottleneck with high-speed reporting



  - Coherent with Iperf
  - Mitigated by in-line filtering

Notes

Notes

List of Figures

List of Tables

Acronyms

3GPP 3^{rd} Generation Partnership Project

ABC Always Best Connected (Gustafsson and Jonsson, 2003)

AIMD additive increase/multiplicative decrease (congestion control)

ANOVA analysis of variance

ANACOLUTHON rhetorical device where a sentence abruptly changes from one structure to another; used to show excitement, confusion, or laziness in, *e.g.*, dramatic monologues or stream of consciousness writing.

API application programming interface

AR access router (MIP/NEMO); the router providing connectivity for a specific access network.

BER bit error rate

BU binding update (MIP/NEMO); procedure by wich an MN registers a new CoA for use with its HA; also refers to the packets sent by the MN during this procedure.

CALM Communications Access for Land Mobiles (ISO 25111:2009)

CCID Congestion Control Identifier; numeric identifier of one of the selectable congestion control mechanisms used by DCCP.

CN correspondent node (MIP/NEMO); network node having established sessions with an MN; may or may no implement mobility features.

COP constrained optimisation problem

CSP constraints satisfaction problemDechter (2003)

CTS clear to send (IEEE Std 802.11-2007); frame sent in response to an RTS request to confirm the channel is free.

CoA care-of address (MIP/NEMO); locator address of an MN to which packets can be delivered from the HA and supporting CNs (RO).

DCCP Datagram Congestion Control Protocol (Kohler *et al.*, 2006b)

DSRC Dedicated Short-Range Communications

DTN Delay-Tolerant Network (Fall, 2003)

ECN Explicit Congestion Notification (Ramakrishnan *et al.*, 2001)

ETT Expected Transmission Time (Draves *et al.*, 2004)

ETX Expected Transmission count (De Couto *et al.*, 2003)

ETSI European Telecommunications Standards Institute

FEC forward error correction

HA home agent (MIP/NEMO); router to an MN's home network in charge of maintaining mobility bindings and forwarding packets for the MN's HoA to its current CoA(s).

HIP Host Identity Protocol (Moskowitz *et al.*, 2008)

HSPA High Speed Packet Access

HoA home address (MIP/NEMO); identifier address of an MN.

I2V infrastructure-to-vehicle communication

IANA Internet Assigned Numbers Authority

ICMP Internet Control Message Protocol; core protocol of the IP suite, used to exchange query or error messages between IP hosts.

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IPPM IP Performance Metrics working group (IETF)

ISO International Organization for Standardization

ITS Intelligent Transportation System

ITU International Telecommunication Union

LTE Long Term Evolution (3GPP)

MAC medium access control

MANET Mobile *Ad hoc* Network

MCoA Multiple Care-of Address (MIP/NEMO)

MIH media-independent handovers (IEEE Std 802.21-2008)

MIP	Mobile IP (Perkins, 2002); often used in this thesis to refer to MIPv6 in unambiguous contexts.
MIPv6	Mobile IPv6 (Johnson *et al.*, 2011)
MNN	mobile network node (NEMO); network node attached to a NEMO; may or may not implement mobility features.
MN	mobile node (MIP/NEMO); network node moving in the network topology while maintaining its established sessions thanks to a mobility protocol.
MOO	multi-objective optimisation
MOS	mean opinion score (ITU-T Recommendation P.800)
MR	mobile router (NEMO); MN acting as the router to a NEMO.
MP	measurement point (OML); group of measurements reported together by an OML-intrumented application.
MS	measurement stream (OML); loose equivalent of an MP within OML's processing and reporting chain.
NAT	network address translation
NFI	no feedback interval (TFRC model, chap. 4); duration of the timer after which expiration TFRC halves its estimate of the receiver's rate; equivalent to Kohler *et al.* (2008)'s *NFT*.
NDT	Network Diagnostic Tool (Zekauskas, 2005)
NEMO	IPv6 NEtwork MObility (Devarapalli *et al.*, 2005); may also refer to a sub-network moving in the network topology while its MR(s) maintain established sessions on behalf of the MNNs.
OMF	cOntrol and Monitoring Framework (Rakotoarivelo *et al.*, 2010)
OML	OMF Measurement Library (White *et al.*, 2010)
OODA	Observe, Orient, Decide, Act (Boyd, 1995); generic concept for an adaptation loop.
OSI	Open System Interconnection (ISO/IEC 7498-1:1994)
PERMANOVA	non-parametric multi-variate analysis of variance (Anderson, 2001)
PSNR	peak signal-to-noise ratio (ANSI T1.TR.74-2001, 2001)
QoE	quality of experience (ITU-T Recommendation P.10/G.100 Amendment 2)
QoS	quality of service

RO route optimisation (MIP/NEMO); extension avoiding the triangular routing problem by allowing a supporting CN to bypass the HA of an MN and directly encaspulate and exchange traffic with its CoAs.

RIR Regional Internet Registry; organisation managing the allocation of Internet numbers (such as IP addresses) within a particular region.

RSS received signal strength

RTS request to send (IEEE Std 802.11-2007); frame sent before a data frame to check the channel availability and reserve it.

RTT round-trip time

SCTP Stream Control Transmission Protocol (Stewart, 2007)

SINR signal-to-interference ratio

SIP Session Initiation Protocol (Rosenberg *et al.*, 2002; Sparks, 2003)

SNMP Simple Network Management Protocol (Harrington *et al.*, 2002)

SNR signal-to-noise ratio

TCP Transmission Control Protocol (Postel, 1981)

TFRC TCP-Friendly Rate Control (Floyd *et al.*, 2008)

UDP User Datagram Protocol (Postel, 1980)

UMTS Universal Mobile Telecommunications System

UPN User-Provided Network (Sofia and Mendes, 2008)

VANET Vehicular *Ad hoc* Network

V2I vehicle-to-infrastructure communication

V2V vehicle-to-vehicle communication

VoD video on demand

VoIP voice over IP

Index

Contributions aux mécanismes de réseau pour un usage adaptatif des ressources mobiles

Résumé: Avec les larges déploiements de multiples technologies sans fil, les terminaux informatiques mobiles bénéficient d'une connectivité presque permanente, changeant de réseaux d'accès au gré de leurs déplacements. Ceci pose cependant le problème de la sélection de ces réseaux, afin de fournir les meilleures performances. Cette mobilité risque aussi d'impacter la qualité des applications, souvent lors des « handovers » d'un réseau à l'autre, ou en raison de la disparité des caractéristiques des réseaux d'accès. Pour aborder ces problèmes, cette thèse introduit et évalue trois éléments de contrôle (observation, décision, action) permettant une meilleure utilisation des ressources réseau par les équipements mobiles.

Nous montrons d'abord qu'un mécanisme de décision qui utilise directement les métriques pertinentes pour les utilisateurs et les applications est plus approprié que l'approche indirecte classique basée sur les métriques réseau. Ce mécanisme contrôle de manière coordonnée l'ensemble de la pile protocolaire, plutôt que des composants séparés, afin d'éviter des combinaisons conflictuelles. Nous démontrons que la flexibilité des paramètres applicatifs peut être exploitée et permet de maintenir une qualité élevée pour les applications tout en réduisant les coûts d'accès (énergétique et financier).

Une extension au *TCP-Friendly Rate Control mechanism* (TFRC) est ensuite introduite, en tant qu'élément d'action, pour atténuer les perturbations lors des handovers. Nous proposons de suspendre la transmission avant la déconnexion, puis de sonder le réseau après reconnexion. Nous montrons que cela permet un rétablissement plus rapide et une meilleure adaptabilité aux conditions du nouveau réseau. Son usage en combinaison avec le *Datagram Congestion Control Protocol* (DCCP) offre un meilleur support aux applications temps-réel, dont la qualité dépend de la vitesse de transmission immédiate.

Finalement, nous présentons une méthode pour évaluer la *OMF Measurement Library* (OML), une bibliothèque d'instrumentation dont nous proposons l'usage comme élément d'observation. Nous montrons que cette bibliothèque n'a pas d'impact significatif sur les applications instrumentées et qu'elle permet un suivi précis des métriques idoines.

Mots clés: mobilité réseau, pile protocolaire, architecture inter-couche, optimisation, qualité d'expérience, protocole de transport

Contributions to Mechanisms for Adaptive Use of Mobile Network Resources

Abstract: With the widespread availability of multiple wireless network technologies, mobile computing devices can benefit from almost uninterrupted connectivity by changing network attachments as they move. This however raises the problem of the selection method to be used for the choice of the wireless networks to associate with, in order to provide the best performance. Moreover, mobility events may result in poor application quality, due to either a disruption in connectivity during the handover or the heterogeneity of the characteristic of different access networks. To address these problems, this thesis introduces and studies all three elements (observation, decision, action) of a control framework to enable better use of available network resources.

We first show that a decision mechanism which directly considers the relevant user- and application-centric metrics is more appropriate than using the common network metrics-based indirect approach. This mechanism is used to control the entire network stack of the mobile node in a coordinated way, rather than individual components, to avoid potentially conflicting combinations. Our results indicate that, by exploiting the flexibility of application parameters, it is possible to maintain high application quality while reducing both the power consumption and access price.

We then introduce a mobility-aware extension to the TCP-Friendly Rate Control mechanism (TFRC), as an action element, to address the disruption in connectivity resulting from the mobility events. We propose to suspend the transmission before disconnections and to probe the network after reconnections. Simulations demonstrate how this enables faster recovery after disconnected periods as well as a significantly improved adaptation to the newly available network conditions. When used with the Datagram Congestion Control Protocol (DCCP), experiments show that it provides better support for real-time applications for which the user-perceived quality is very dependent on the immediate transmission rate.

Finally, we present an experimental process to evaluate the OMF Measurement Library (OML), a lightweight instrumentation and reporting tool which we propose to use as the observation element of our framework. We show that this library does not significantly impact the performance of the instrumented applications, while accurately reporting the observed metrics.

Keywords: network mobility, communication stack, cross-layer framework, optimisation, quality of experience, transport protocol

MINES ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY